Optimization

Optimization

$$\begin{cases} \min\limits_{x} \ \boxed{f(x)} \text{ — objective function} \\ \text{s.t.} \quad g(x) = 0 \longrightarrow \text{equality constraints} \\ \qquad\ \ h(x) \geq 0 \longrightarrow \text{inequality constraints} \end{cases}$$

$f(x,y) = \sin x \sin y$
$-\pi \leq x, y \leq \pi$

* Unconstrained optimization :

1st order necessary conditions : $f'(x^*) = 0$ $\qquad \nabla f(\underset{\sim}{x}^*) = \underset{\sim}{0}$

$\Longrightarrow x^*$ is a critical point

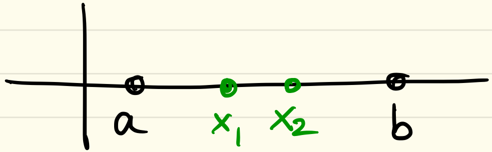
2nd order sufficient conditions $\quad f''(x^*) > 0 \quad H_f(x^*)$ pos. def MIN.

$f''(x^*) < 0 \quad H_f(x^*)$ neg. def MAX.

$f''(x^*) = 0 \quad H_f(x^*)$ indefinite SADDLE

Optimization Methods to solve 1D problems

☆ Golden Section Search



- interval $(a,b) \Rightarrow h_0 = b-a$
- $h_{k+1} = h_k \tau \qquad \tau = 0.618$
- points inside the interval: $x_1 = a + (1-\tau)h_k$
  $x_2 = a + \tau h_k$
- linear convergence $\lim\limits_{k\to\infty} \frac{e_{k+1}}{e_k} = 0.618$
- Require only $\underline{\underline{1}}$ fc. eval per iteration / $\underline{\underline{\text{No}}}$ need for derivatives!

☆ Newton

- initial guess x_0
- update $x_{k+1} = x_k - f'(x_k)/f''(x_k)$
- requires $f'(x)$ and $f''(x)$
- quadratic convergence (local)
- depends on initial guess

Optimization in ND: Steepest Descent Method

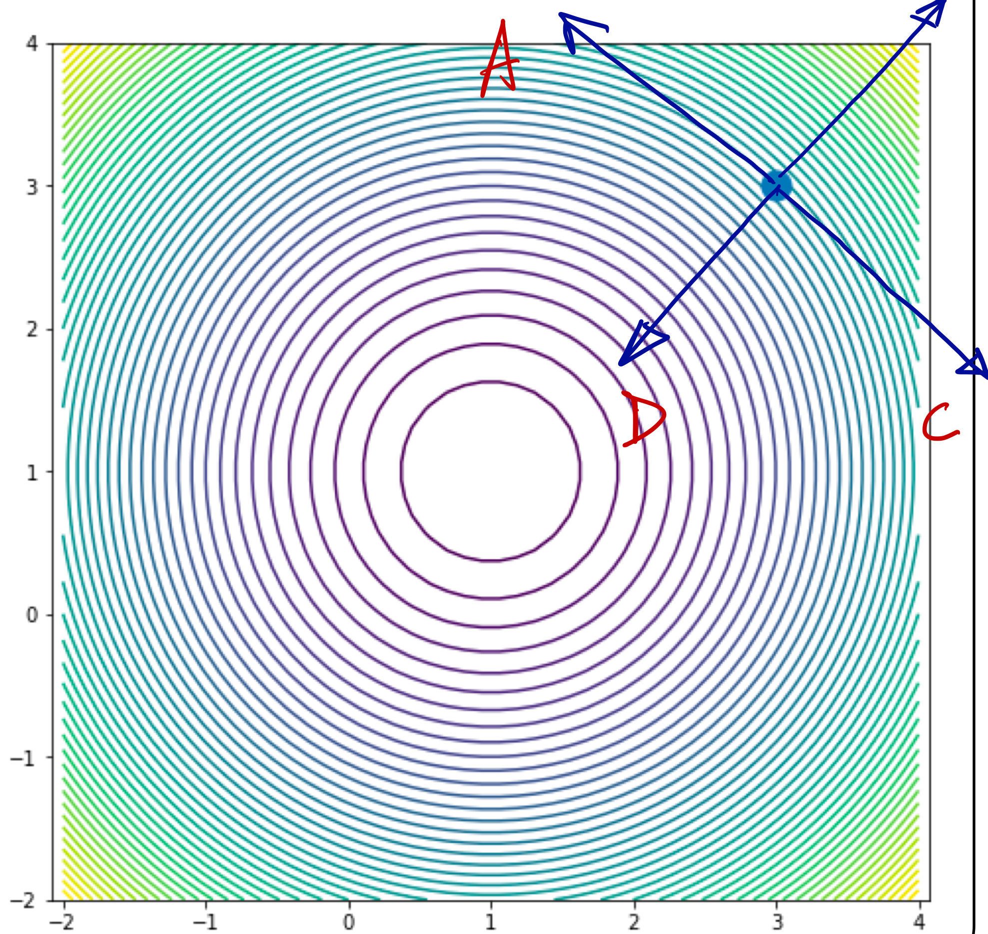
Given a function $f(\boldsymbol{x}): \mathcal{R}^n \rightarrow \mathcal{R}$ at a point $\boldsymbol{x}$, the function will decrease its value in the direction of steepest descent: $-\boldsymbol{\nabla} f(\boldsymbol{x})$

Iclicker question:

What is the steepest descent direction?

D

$$f(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^2$$



Steepest Descent Method

Start with initial guess:

$$\boldsymbol{x}_0 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

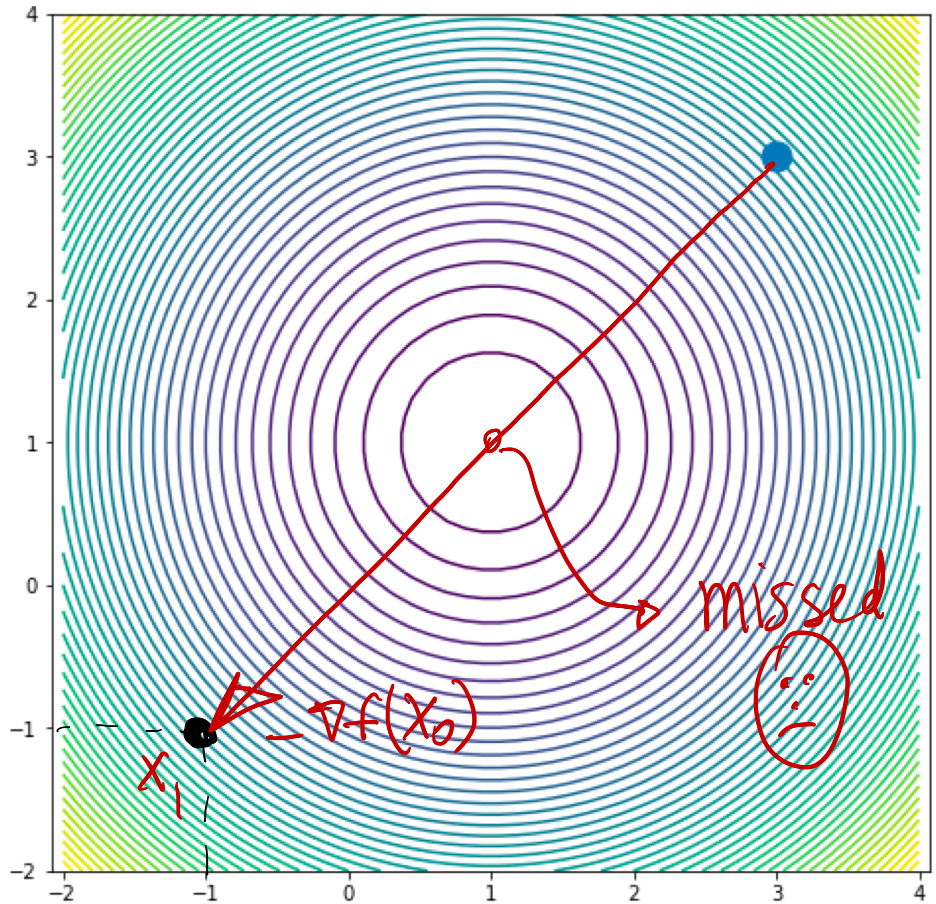
Check the update:

$$\underset{\sim}{x}_1 = \underset{\sim}{x}_0 - \nabla f(x_0)$$

$$\nabla f = \begin{bmatrix} 2(x_1 - 1) \\ 2(x_2 - 1) \end{bmatrix} \Rightarrow \nabla f(x_0) = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$$

$$\underset{\sim}{x}_1 = \begin{bmatrix} 3 \\ 3 \end{bmatrix} - \begin{bmatrix} 4 \\ 4 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

$$f(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^2$$



Steepest Descent Method

Update the variable with:

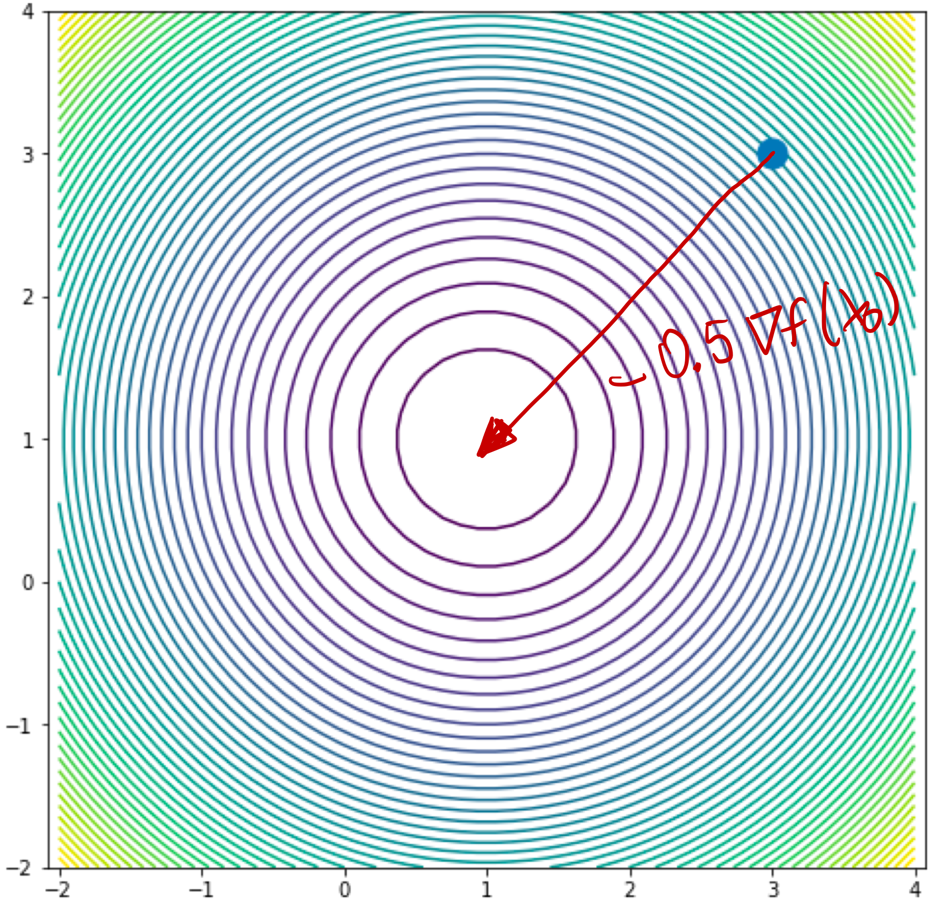
$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \boldsymbol{\nabla} f(\boldsymbol{x}_k)$$

How far along the gradient should we go? What is the "best size" for α_k?

A) 0
B) $\boxed{0.5}$
C) 1
D) 2
E) Cannot be determined

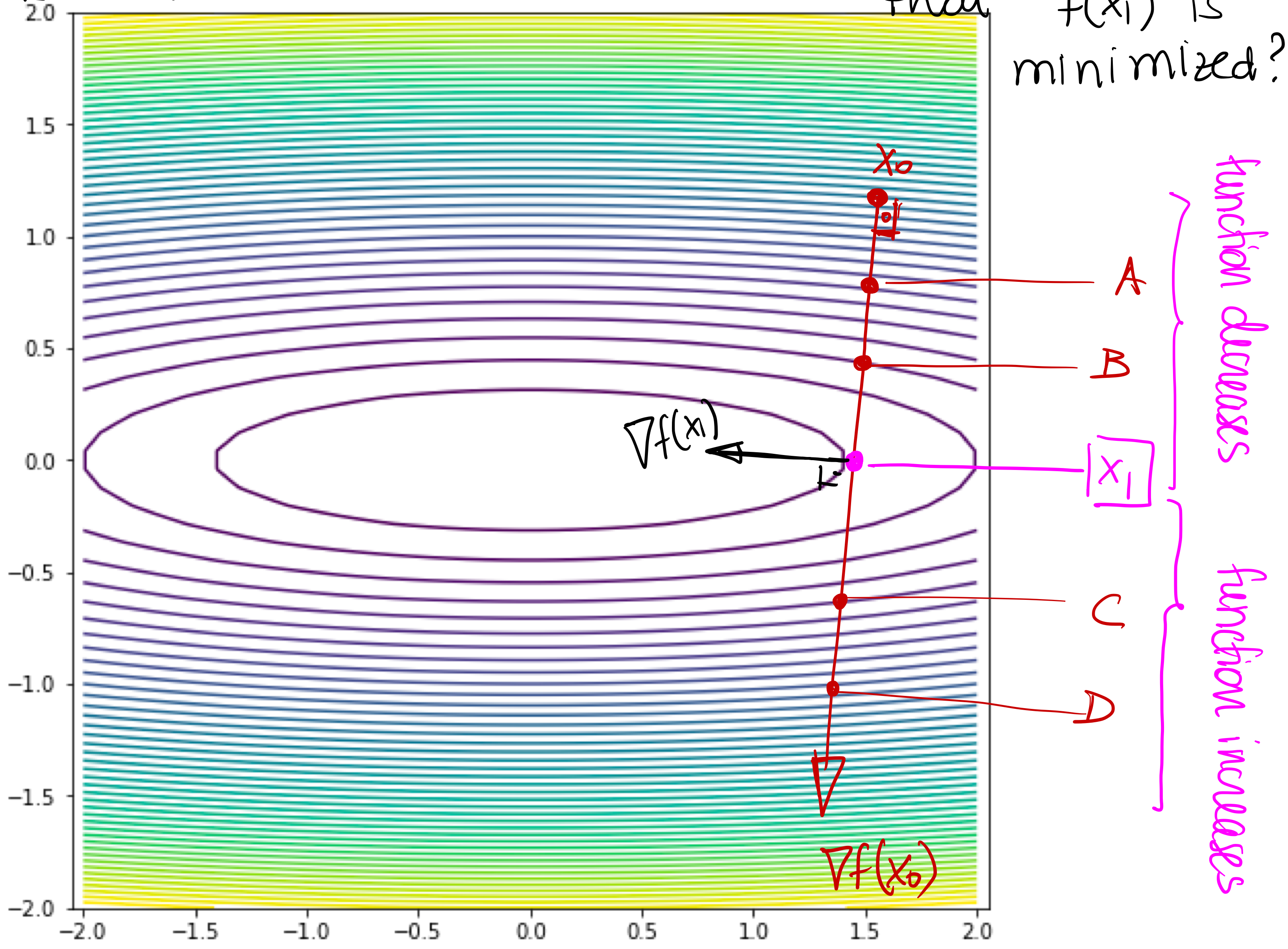
What is the best α_k ?

$$f(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^2$$



$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) \longrightarrow x_1 = x_0 - \alpha \nabla f(x_0)$$

What is α such that $f(x_1)$ is minimized?



Steepest Descent Method

★ Need $f(\underset{\sim}{x})$ and $\nabla f(\underset{\sim}{x})$

★ "Many" function evals inside line search

Algorithm:

Initial guess: $\boldsymbol{x}_0$

Evaluate: $\boldsymbol{s}_k = -\boldsymbol{\nabla} f(\boldsymbol{x}_k)$

Perform a line search to obtain α_k (for example, Golden Section Search)

$$\alpha_k = \underset{\alpha}{\operatorname{argmin}}\, f(\boldsymbol{x}_k + \alpha\, \boldsymbol{s}_k)$$

fixed

fixed

Update: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\, \boldsymbol{s}_k$

Line Search

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

what is α_k such that $f(x_{k+1})$ is minimized?

$$\min_{\alpha} f(\underbrace{x_k - \alpha \nabla f(x_k)}_{x_{k+1}})$$

Necessary condition: $\frac{df}{d\alpha} = 0$

Chain rule:

$$\frac{df}{d\alpha} = \frac{df}{dx_{k+1}} \frac{dx_{k+1}}{d\alpha} \qquad \frac{dx_{k+1}}{d\alpha} = -\nabla f(x_k)$$

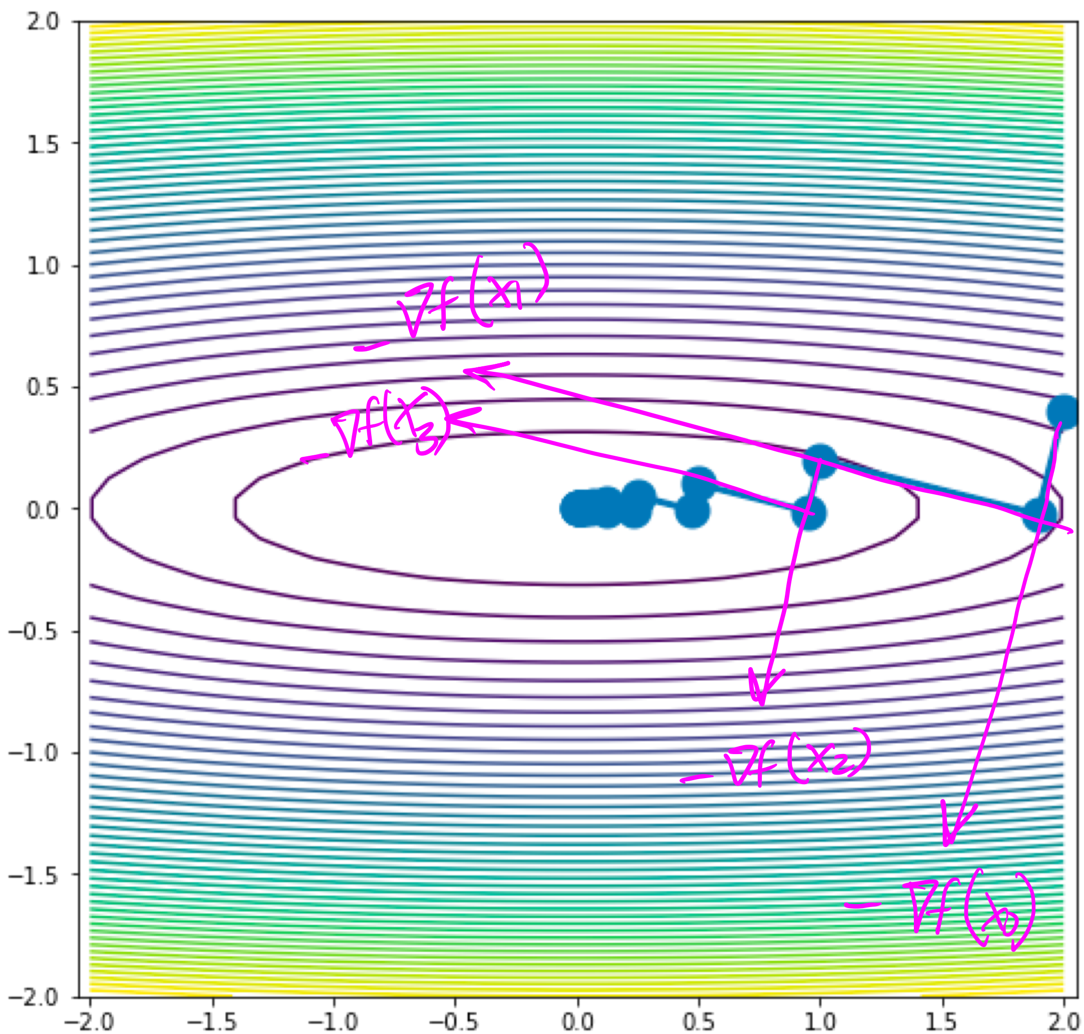
$$= -\nabla f(x_{k+1})^T \nabla f(x_k) = 0$$

$\nabla f(x_{k+1})$ is orthogonal to $\nabla f(x_k)$!

Steepest Descent Method

Demo: Steepest Descent

Convergence: linear



Demo: "Steepest Descent"

Iclicker question:

Consider minimizing the function

$$f(x_1, x_2) = 10(x_1)^3 - (x_2)^2 + x_1 - 1$$

Given the initial guess

$$x_1 = 2,\ x_2 = 2$$

what is the direction of the first step of gradient descent?

$$\nabla f = \begin{bmatrix} 30x_1^2 + 1 \\ -2x_2 \end{bmatrix} \longrightarrow \nabla f(2,2) = \begin{bmatrix} 121 \\ -4 \end{bmatrix}$$

direction of gradient descent $\longrightarrow \begin{bmatrix} -121 \\ +4 \end{bmatrix}$

Newton's Method

★ Taylor expansion:

$$f(\underset{\sim}{x}+\underset{\sim}{s}) \cong \underbrace{f(\underset{\sim}{x}) + \underset{\sim}{\nabla} f(\underset{\sim}{x})^T \underset{\sim}{s} + \frac{1}{2} \underset{\sim}{s}^T \underset{=}{H}(\underset{\sim}{x}) \underset{\sim}{s}}_{\text{quadratic approximation of } f(\underset{\sim}{x})} = \tilde{f}(\underset{\sim}{s})$$

★ First-order necessary condition:

$$\underset{\sim}{\nabla} \hat{f}(\underset{\sim}{s}) = \underset{\sim}{\nabla} f + \frac{1}{2}\left(\underset{=}{H}\underset{\sim}{s} + \underset{=}{H}\underset{\sim}{s}\right) = \underset{\sim}{0}$$

$$\boxed{\underset{\sim}{H}\,\underset{\sim}{s} = -\underset{\sim}{\nabla} f}$$

→ solve linear system of equations for $\underset{\sim}{s}$!

Newton's Method

Algorithm:

Initial guess: $\boldsymbol{x}_0$

Solve: $\boldsymbol{H}_{\boldsymbol{f}}(\boldsymbol{x}_k)\,\boldsymbol{s}_k = -\boldsymbol{\nabla} f(\boldsymbol{x}_k)$

Update: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k$

Note that the Hessian is related to the curvature and therefore contains the information about how large the step should be. (No need for line search!)

Iclicker question

To find a minimum of the function $f(x, y) = 3x^2 + 2y^2$, which is the expression for one step of Newton's method?

$\nabla f = \begin{bmatrix} 6x \\ 4y \end{bmatrix}$

A) $\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - \begin{bmatrix} 6 & 0 \\ 0 & 4 \end{bmatrix}^{-1} \begin{bmatrix} 6x_k \\ 4y_k \end{bmatrix}$

$H_f = \begin{bmatrix} 6 & 0 \\ 0 & 4 \end{bmatrix}$

B) $\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = -\begin{bmatrix} 6 & 0 \\ 0 & 4 \end{bmatrix}^{-1} \begin{bmatrix} 6x_k \\ 4y_k \end{bmatrix}$

$H s = -\nabla f$

C) $\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} 6 & 0 \\ 0 & 4 \end{bmatrix}^{T} \begin{bmatrix} 6x_k \\ 4y_k \end{bmatrix}$

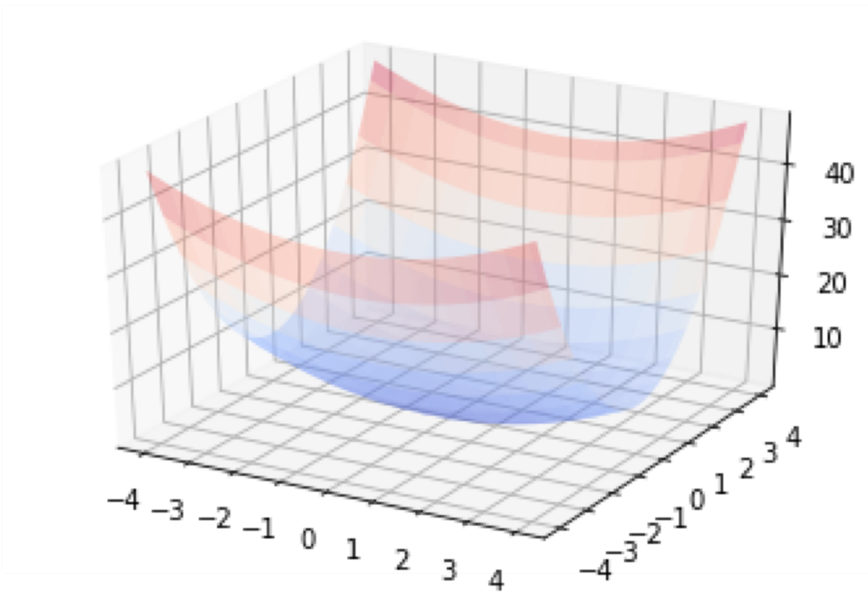
$\begin{bmatrix} 6 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} -6x \\ -4y \end{bmatrix}$

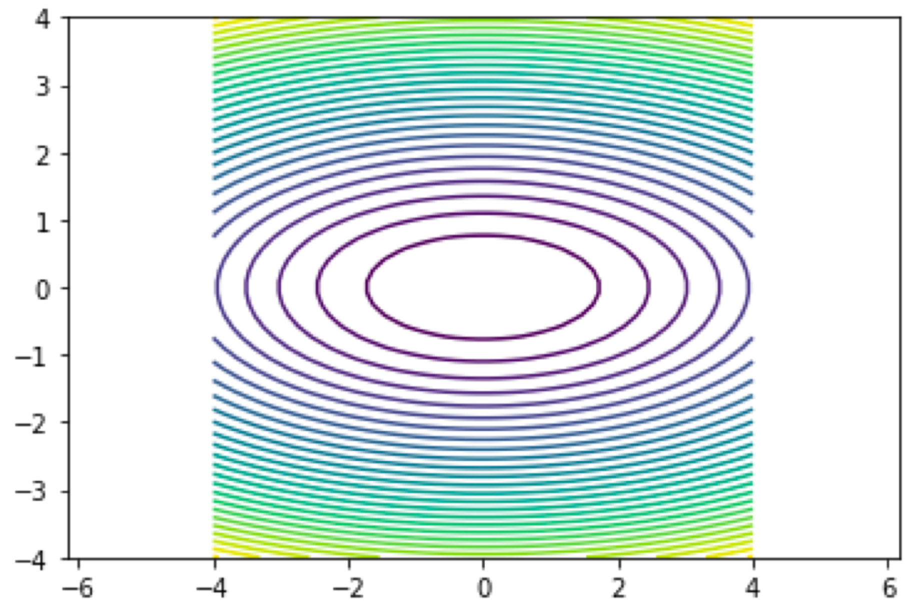
D) $\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - \begin{bmatrix} 6 & 0 \\ 0 & 4 \end{bmatrix}^{T} \begin{bmatrix} 6x_k \\ 4y_k \end{bmatrix}$

$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - \begin{bmatrix} 6 & 0 \\ 0 & 4 \end{bmatrix}^{-1} \begin{bmatrix} 6x_k \\ 4y_k \end{bmatrix}$

Iclicker question:

$$f(x, y) = 0.5x^2 + 2.5y^2$$





When using the Newton's Method to find the minimizer of this function, estimate the number of iterations it would take for convergence?

A) 1 B) 2-5 C) 5-10 D) More than 10 E) Depends on the initial guess

Newton's Method Summary

Algorithm:

Initial guess: $\boldsymbol{x}_0$

Solve: $\boldsymbol{H}_{\boldsymbol{f}}(\boldsymbol{x}_k)\,\boldsymbol{s}_k = -\boldsymbol{\nabla} f(\boldsymbol{x}_k)$

Update: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k$

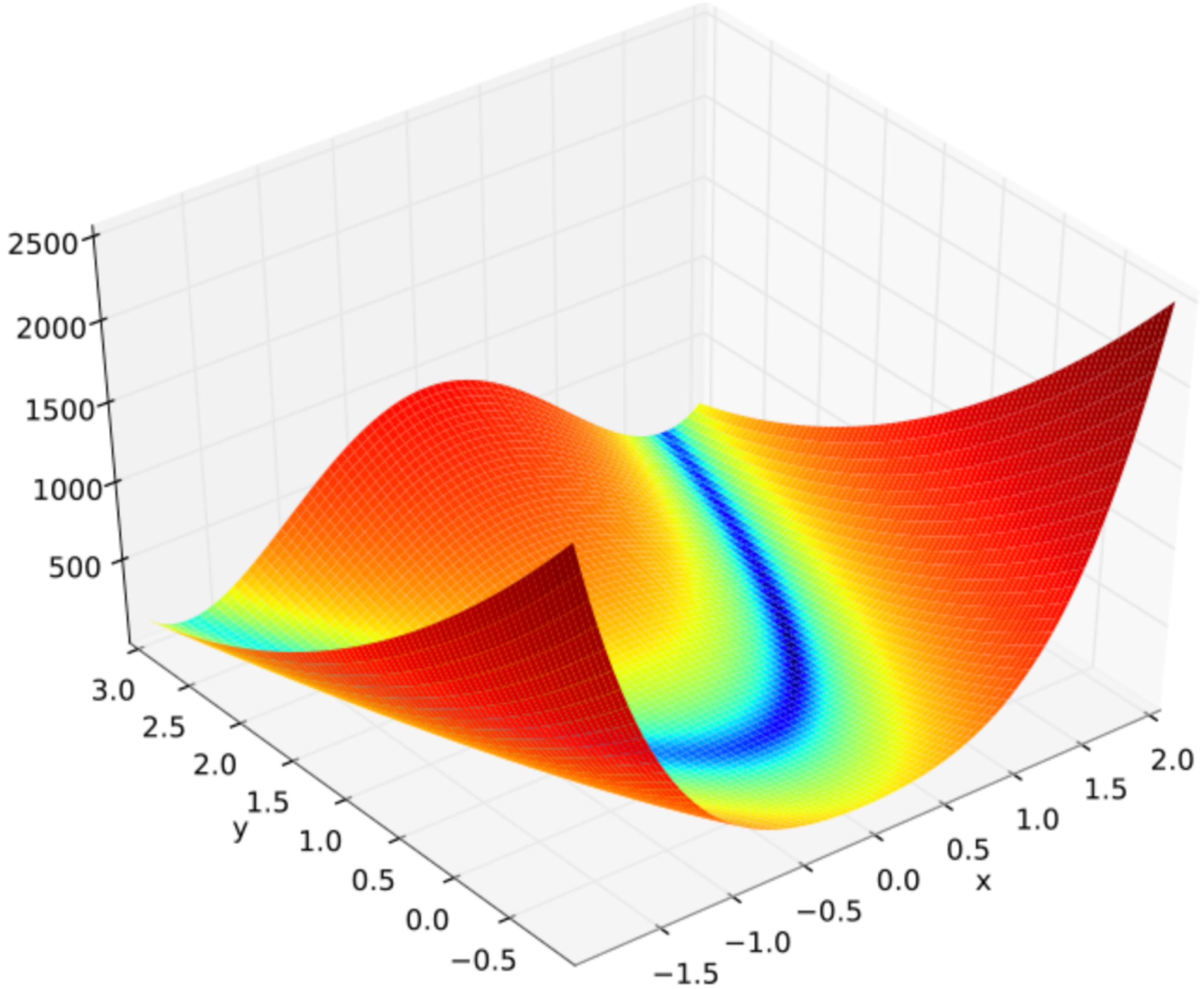
About the method…

- Typical quadratic convergence ☺
- Need second derivatives ☹
- Local convergence (start guess close to solution)
- Works poorly when Hessian is nearly indefinite
- Cost per iteration: $O(n^3)$

Demo: ”Newton's method in n dimensions”

Example:

https://en.wikipedia.org/wiki/Rosenbrock_function



Iclicker question:

Recall Newton's method and the steepest descent method for minimizing a function $f(\boldsymbol{x}): \mathcal{R}^n \rightarrow \mathcal{R}$. How many statements below describe the Newton Method's only (not both)?

1. Convergence is linear SD
2. Requires a line search at each iteration SD
3. Evaluates the Gradient of $f(\boldsymbol{x})$ at each iteration Both
4. Evaluates the Hessian of $f(\boldsymbol{x})$ at each iteration Newton
5. Computational cost per iteration is $O(n^3)$ Newton

A) 1 $\boxed{\text{B) 2}}$ C) 3 D) 4 E) 5