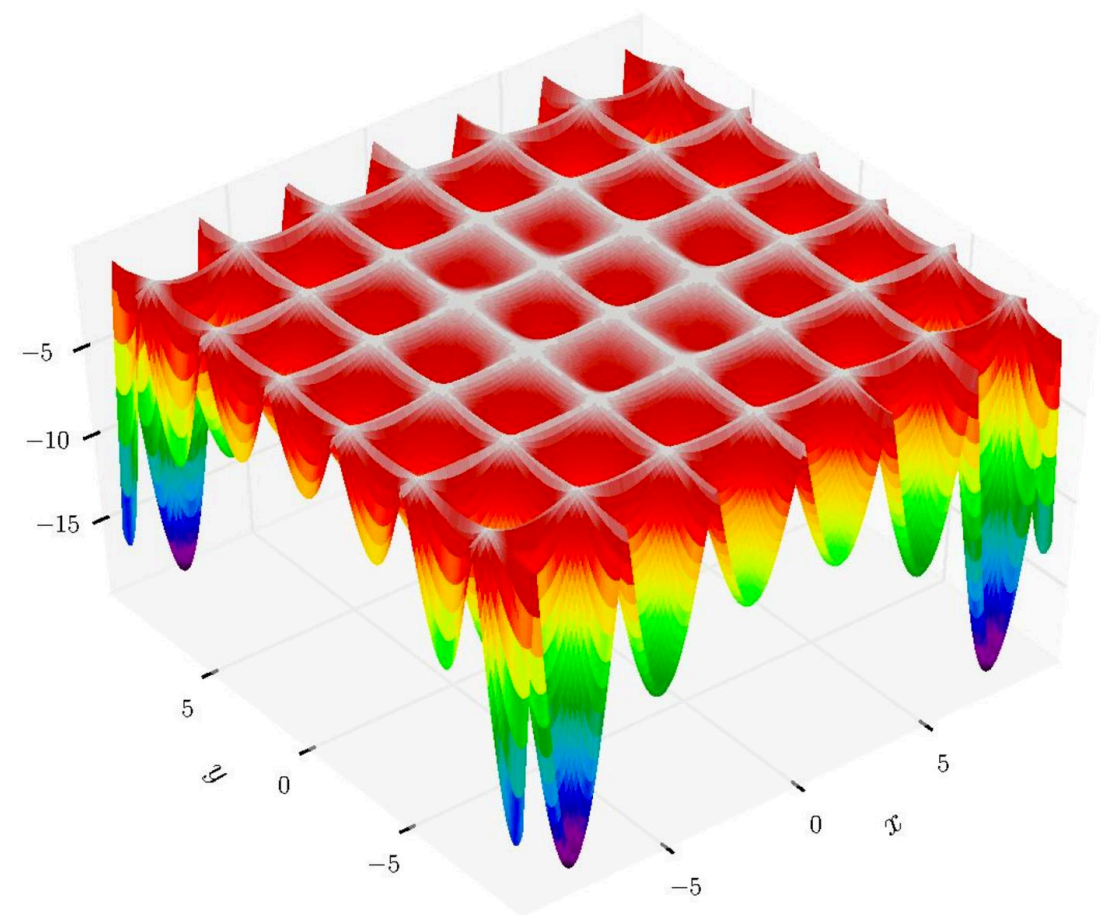
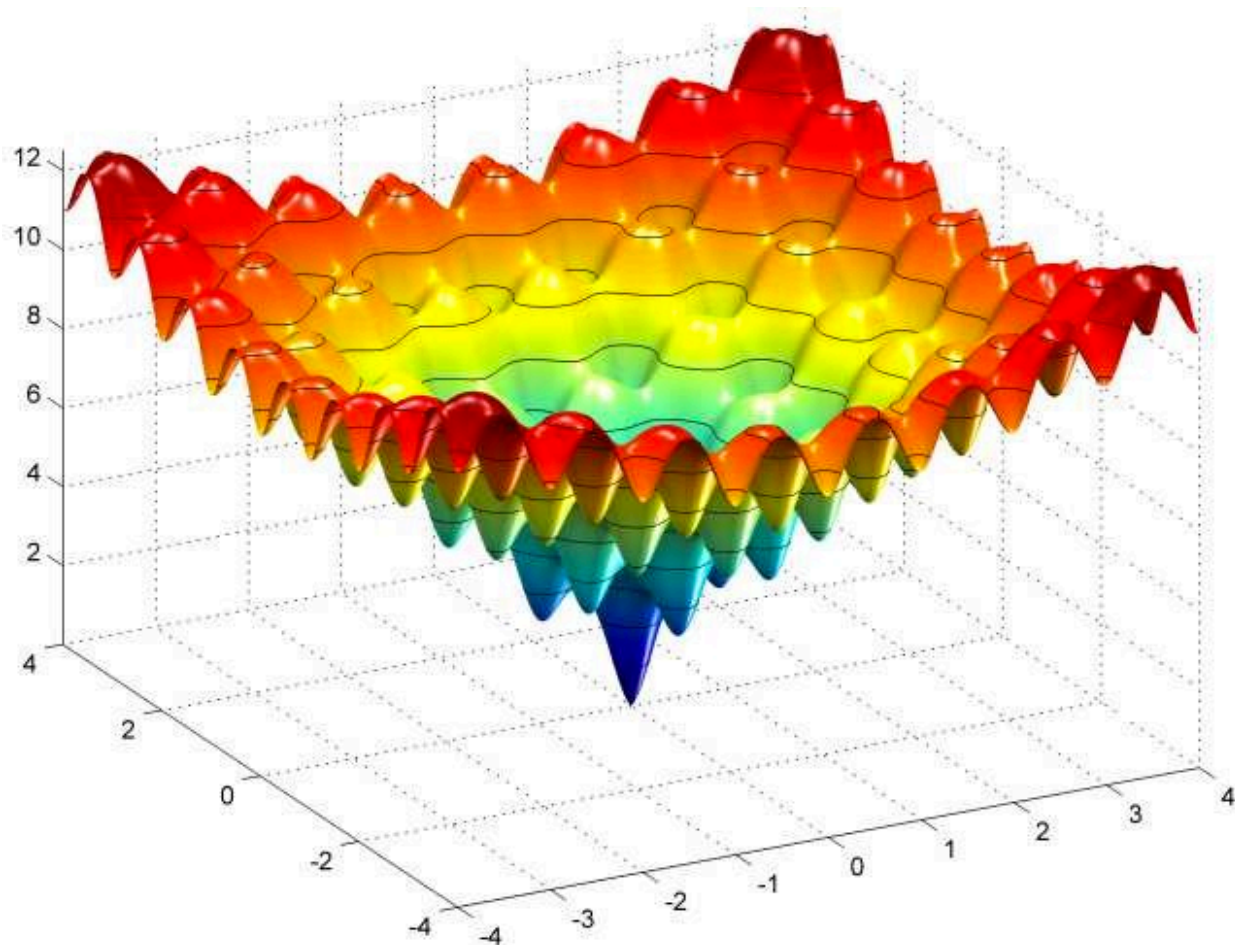


# Numerical Optimization I

UCSD CSE 257

Sicun Gao

# Global vs. Local Minima



# Local Minima

- $f : \mathbb{R}^n \rightarrow \mathbb{R}$

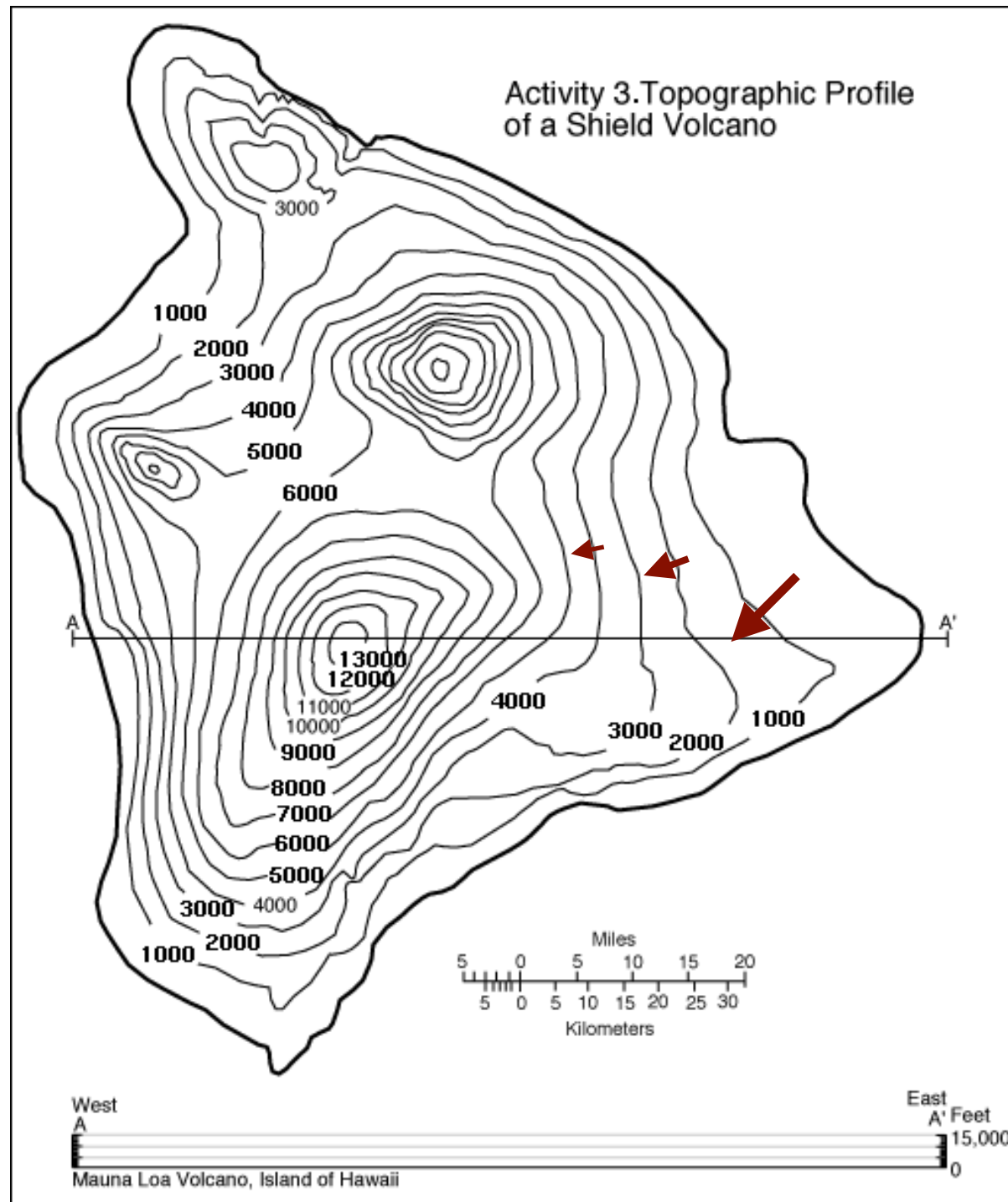
- Weak local minimizer:

$$\exists \varepsilon \forall y \left( y \in B(x, \varepsilon) \rightarrow f(x) \leq f(y) \right)$$

- Strict/Strong local minimizer:

$$\exists \varepsilon \forall y \left( y \in B(x, \varepsilon) \wedge x \neq y \rightarrow f(x) < f(y) \right)$$

# Level Sets and Gradients



Level sets:

$$f(x) = c$$

Gradient (if differentiable):

$$\nabla f(x) = \left( \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right)^T$$

Why always orthogonal?

# First-order Taylor Expansion

$$f(x + p) = f(x) + \nabla f(x + tp)^T p$$

for some  $t \in (0,1)$

$$= f(x) + \nabla f(x)^T p + O(\|p\|_2^2)$$

- First-order condition for convexity?

# First-order Necessary Condition

- If  $x^*$  is a local minimizer and  $f$  is  $C^1$  in an open neighborhood of  $x^*$ , then

$$\nabla f(x^*) = 0$$

# First-order Necessary Condition

- If  $x^*$  is a local minimizer and  $f$  is  $C^1$  in an open neighborhood of  $x^*$ , then

$$\nabla f(x^*) = 0$$

- Proof: Suppose not. Consider the direction  $p = -\nabla f(x^*)$ ,

$$\nabla f(x^*)^T p = -\|\nabla f(x^*)\|^2 < 0$$

There exists some  $\hat{t}$  such that for all  $t \in [0, \hat{t}]$ ,  $\nabla f(x^* + tp)^T p < 0$ .  
Then consider all points in that range in that direction.

Where is continuity used?

# Second-order Taylor Expansion

- Hessian

$$(\nabla^2 f)_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

- Second-order

$$f(x + p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x + tp) p$$

$t \in (0,1)$

- Second-order condition for convexity?

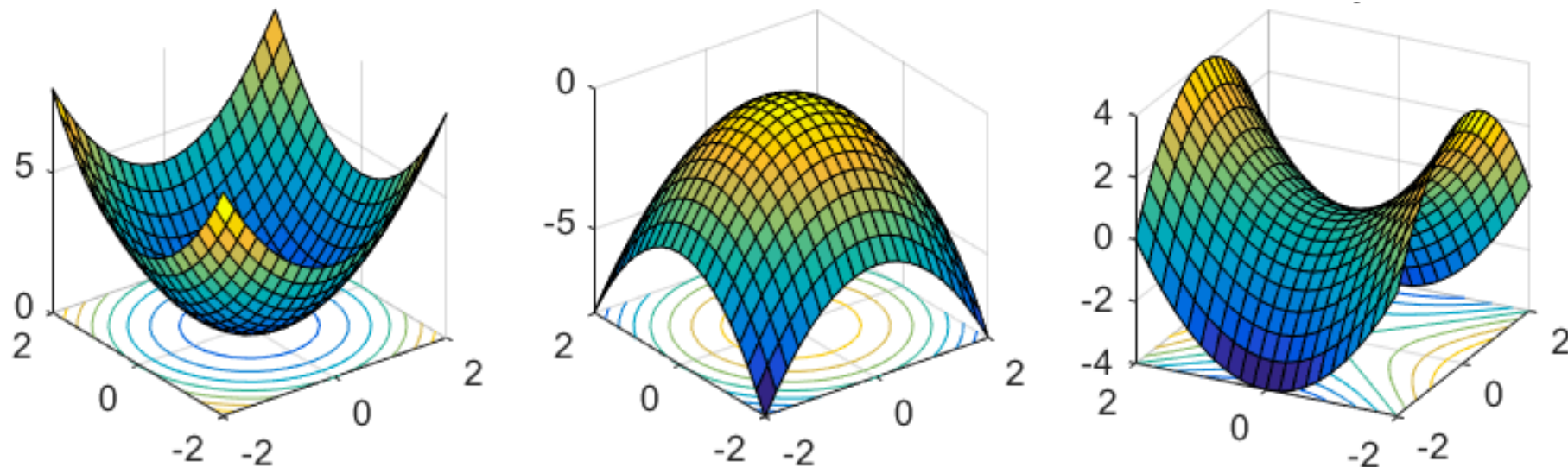


# Critical points

- When the first-order necessary condition is satisfied,

$$\nabla f(x^*) = 0$$

$x^*$  is called a critical/stationary point. Different types:



# Second-order Necessary Condition

- If  $x^*$  is a local minimizer and  $f$  is  $C^2$  in an open neighborhood of  $x^*$ , then

$$\nabla f(x^*) = 0 \text{ and } \nabla^2 f(x^*) \succeq 0$$

# Second-order Necessary Condition

- If  $x^*$  is a local minimizer and  $f$  is  $C^2$  in an open neighborhood of  $x^*$ , then

$$\nabla f(x^*) = 0 \text{ and } \nabla^2 f(x^*) \succeq 0$$

- Proof: Suppose not. Consider  $p$  s.t.  $p^T \nabla^2 f(x)p < 0$ . There exists a small enough neighborhood around  $x^*$  such that  $p^T \nabla^2 f(x + tp)p < 0$  for all  $t \in [0, \hat{t}]$ . Consider all points in that range in that direction and use Taylor expansion.

# Second-order Sufficient Condition

- If  $f$  is in  $C^2$  in an open neighborhood of  $x^*$ , and

$$\nabla f(x^*) = 0 \text{ and } \nabla^2 f(x^*) \succ 0$$

then  $x^*$  is a **strict** local minimizer of  $f$ .

# Second-order Sufficient Condition

- If  $f$  is in  $C^2$  in an open neighborhood of  $x^*$ , and

$$\nabla f(x^*) = 0 \text{ and } \nabla^2 f(x^*) \succ 0$$

then  $x^*$  is a **strict** local minimizer of  $f$ .

- Proof: Choose  $r \in \mathbb{R}^+$  such that  $\nabla^2 f(z) \succ 0$  for all  $z \in B(x^*, r)$ . Consider any nonzero vector  $\|p\| < r$ . Apply Taylor expansion.

only sufficient, not necessary for strict minimizer

# Descent Methods

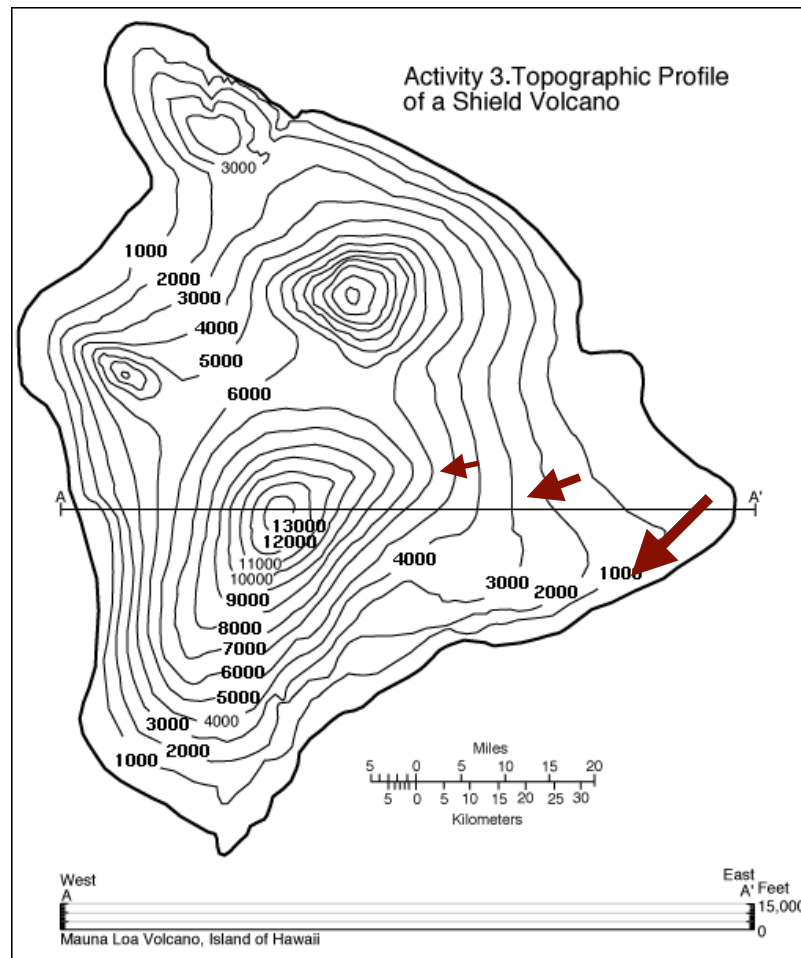
- Input:  $x_0 \in \text{dom}(f)$
- while not **stopping criterion**
  - $p_k \leftarrow$  Choose a **descent direction**
  - $\alpha_k \leftarrow$  Choose a **step size**
  - $x_{k+1} \leftarrow x_k + \alpha_k p_k$

# Gradient Descent

Gradient:

$$\nabla f(x) = \left( \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right)^T$$

Common issue:



# Newton Direction

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

Take derivative over  $p$

$$\frac{\partial f(x + p)}{\partial p} = \nabla f(x)^T + p^T \nabla^2 f(x)$$



# Newton Direction

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

Take derivative over  $p$

$$\frac{\partial f(x + p)}{\partial p} = \nabla f(x)^T + p^T \nabla^2 f(x)$$

Setting the derivative to 0 and assume  $\nabla^2 f \succ 0$

$$p = - (\nabla^2 f(x))^{-1} \nabla f(x)$$

# Newton Direction

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

Setting the derivative to 0 and assume  $\nabla^2 f \succ 0$

$$p = - (\nabla^2 f(x))^{-1} \nabla f(x)$$

Such  $p$  is a descent direction:

$$\nabla f^T p = - \nabla f^T (\nabla^2 f)^{-1} \nabla f < 0$$

So it can only be used as a line search direction when  $\nabla^2 f \succ 0$ .

# Line Search

In each iteration, need to determine two things

- Search direction  $p_k$ 
  - descent direction: any direction with  $\nabla f(x_k)^T p_k < 0$
- How far to move along that direction  $\alpha$ 
  - learning rate

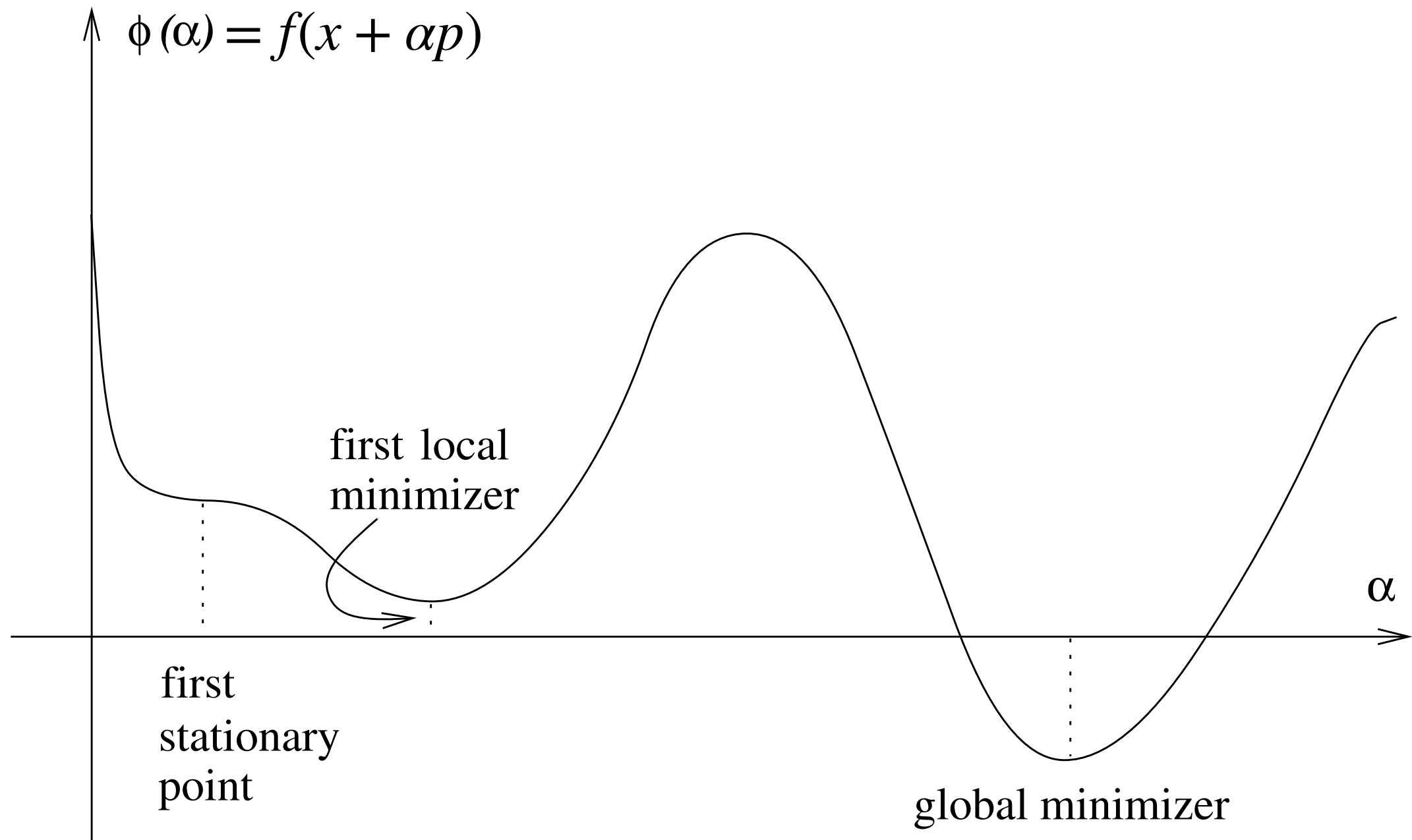
# Exact Line Search

Along the direction of choice  $p$ , choose step size that maximally reduces  $f(x + \alpha p)$  in each step

$$\alpha \leftarrow \arg \min_{\alpha \geq 0} f(x + \alpha p)$$

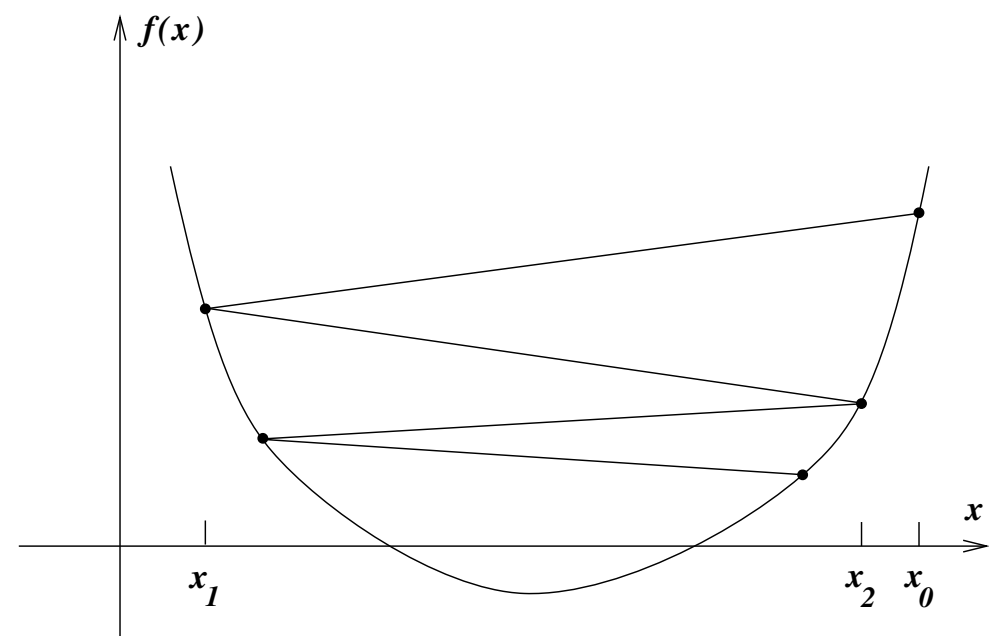
- It reduces n-dimensional to one-dimensional problem
- Exact solving is too costly
- Is exact solving always the best anyway?

# Inexact Line Search



# Sufficient Decrease

- Suppose we can not afford exact line search, then we need some condition for choosing  $\alpha$  in each step
- The simple condition  $f(x_{k+1}) = f(x_k + \alpha_k p_k) < f(x_k)$  does not guarantee convergence



# Sufficient Decrease: Armijo Rule

- Use a step size that sees “reasonable decrease” according to relaxed first-order model

$$f(x + \alpha p) \leq f(x) + \alpha c \nabla f(x)^T p$$

for some fixed  $c \in (0,1)$

- The constant  $c$  is important, since whenever the function is convex,

$$f(x + \alpha p) \geq f(x) + \alpha \nabla f(x)^T p$$

# Backtracking Line Search

- Start with some arbitrary  $\alpha$  and iteratively decrease until Armijo rule is satisfied

**repeat** until  $f(x_k + \alpha p_k) \leq f(x_k) + c\alpha \nabla f_k^T p_k$

$\alpha \leftarrow \rho\alpha;$

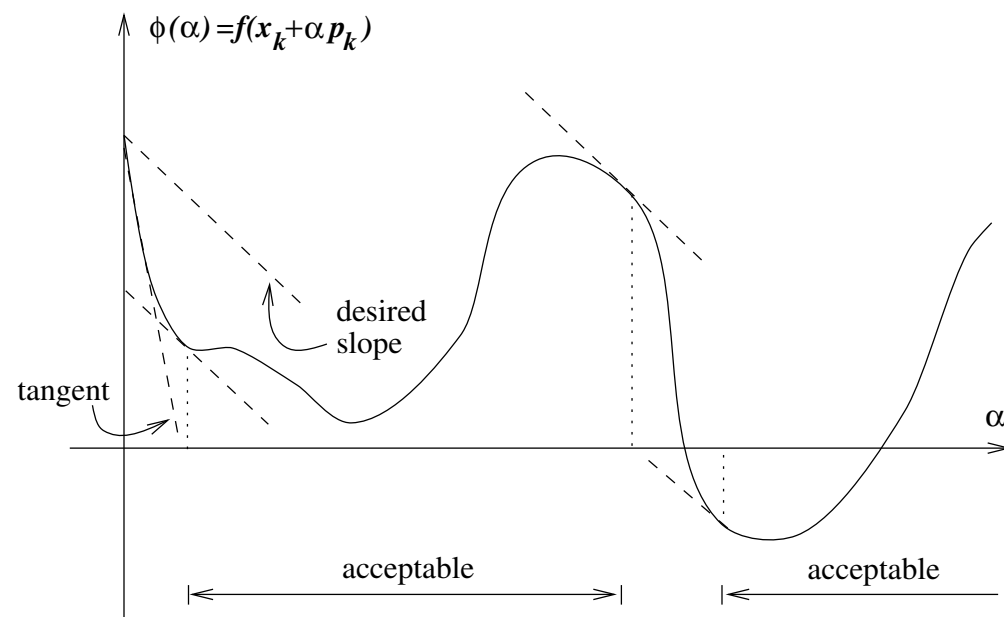
**end (repeat)**

for some fixed  $\rho \in (0,1)$



# Extra: Curvature Condition

- For more general analysis (when not using specific algorithms like backtracking line search), we also need to ensure that the steps are not too short
- Stop only when descent along  $\alpha$  has slowed down



$$\nabla f_{k+1}^T p_k \geq c' \nabla f_k^T p_k$$

$$c' \in (c, 1)$$

# Wolfe Conditions/Steps

- The Armijo rule and the curvature condition put together are called the Wolfe Conditions
- Stepsizes that satisfy the Wolfe conditions always exist, and they give very general convergence proofs for iterative descent methods

# Existence of Wolfe Steps

Assume  $f$  is lower-bounded (so that minimization is well-defined).

- Sufficient decrease: Consider the difference between the function and the first-order model

$$f_k + c_1 \alpha_k \nabla f_k^T p_k - f(x_k + \alpha_k p_k)$$

It is positive when  $\alpha_k$  is close to zero, and negative as  $\alpha_k$  goes far. So there must be a smallest  $\alpha^*$  such that

$$f_k + c_1 \alpha^* \nabla f_k^T p_k = f(x_k + \alpha^* p_k)$$

# Existence of Wolfe Steps

- Curvature condition: Assume

$$f_k + c_1 \alpha^* \nabla f_k^T p_k = f(x_k + \alpha^* p_k)$$

There exists  $\bar{\alpha} \in (0, \alpha^*)$  such that (Taylor)

$$f(x_k + \alpha^* p_k) = f(x_k) + \nabla f(x_k + \bar{\alpha} p_k)^T \cdot \alpha^* p_k$$

Linking the two equations,

$$\nabla f(x_k + \bar{\alpha} p_k)^T p_k = c_1 \nabla f_k^T p_k > c_2 \nabla f_k^T p_k$$

Since  $c_2 < c_1$  and  $p_k$  is a descent direction.

# Convergence under Wolfe Conditions

- Suppose the gradient of  $f$  is Lipschitz-continuous over an open set  $S$  that contains  $\{x \mid f(x) \leq f(x_0)\}$ , i.e., for some  $L > 0$

$$\forall x, x' \in S \quad \|\nabla f(x) - \nabla f(x')\| \leq L\|x - x'\|$$

Let  $p_k$  be the descent direction in each iteration and  $\alpha_k$  be the step size that satisfies Wolfe Conditions. Then

$$\sum_{k=0}^{\infty} \cos^2 \theta_k \|\nabla f_k\|^2 < \infty$$

$$\cos \theta_k = \frac{-\nabla f_k p_k}{\|\nabla f_k\| \|p_k\|}$$

# Convergence under Wolfe Conditions

$$\sum_{k=0}^{\infty} \cos^2 \theta_k \|\nabla f_k\|^2 < \infty$$

- Usefulness of the convergence:

- $\cos^2 \theta_k \|\nabla f_k\|^2 \rightarrow 0$  so as long as

$$\cos \theta_k \geq \delta > 0 \quad (\text{meaning?})$$

the sequence must converge to  $\nabla f_k \rightarrow 0$

- In fact, as long as  $\cos \theta$  is periodically lower-bounded.

# Convergence under Wolfe Conditions

- Proof sketch:

First show that the step  $\alpha_k$  is lower-bounded

$$\alpha_k \geq C \cdot \frac{-\nabla f^T p_k}{\|p_k\|^2}$$

which allows enough reduction in  $f$  in each iteration

$$f_{k+1} \leq f_k - C' \cdot \frac{(\nabla f^T p_k)^2}{\|p_k\|^2}$$

which turns into  $f_{k+1} \leq f_k - C' \cdot \cos^2 \theta_k \|\nabla f_k\|^2$

# Convergence Rate

- On strongly convex quadratic functions and exact line search, steepest descent satisfies

$$\|x_{k+1} - x^*\|_Q^2 \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 \|x_k - x^*\|_Q^2$$

$0 < \lambda_1 \leq \dots \leq \lambda_n$  are the eigenvalues of the Hessian  $Q$

- For general (non-quadratic but with p.d. Hessian) the rate is  $r^2$  for some  $r \in (\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}, 1)$  when  $k$  is sufficiently large.



# Convergence Rate

- On strongly convex quadratic functions and exact line search, steepest descent satisfies

$$\|x_{k+1} - x^*\|_Q^2 \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 \|x_k - x^*\|_Q^2$$

- For general (non-quadratic but with p.d. Hessian) the rate is  $r^2$  for some  $r \in (\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}, 1)$  when  $k$  is sufficiently large.
- Newton directions under the same assumptions allow quadratic convergence.

# Conjugate Gradient

- For quadratic problems we can do better than using generic directions
- Conjugate gradients allow us to minimize convex quadratic objectives over  $\mathbb{R}^n$

$$f(x) = x^T Q x - b^T x$$

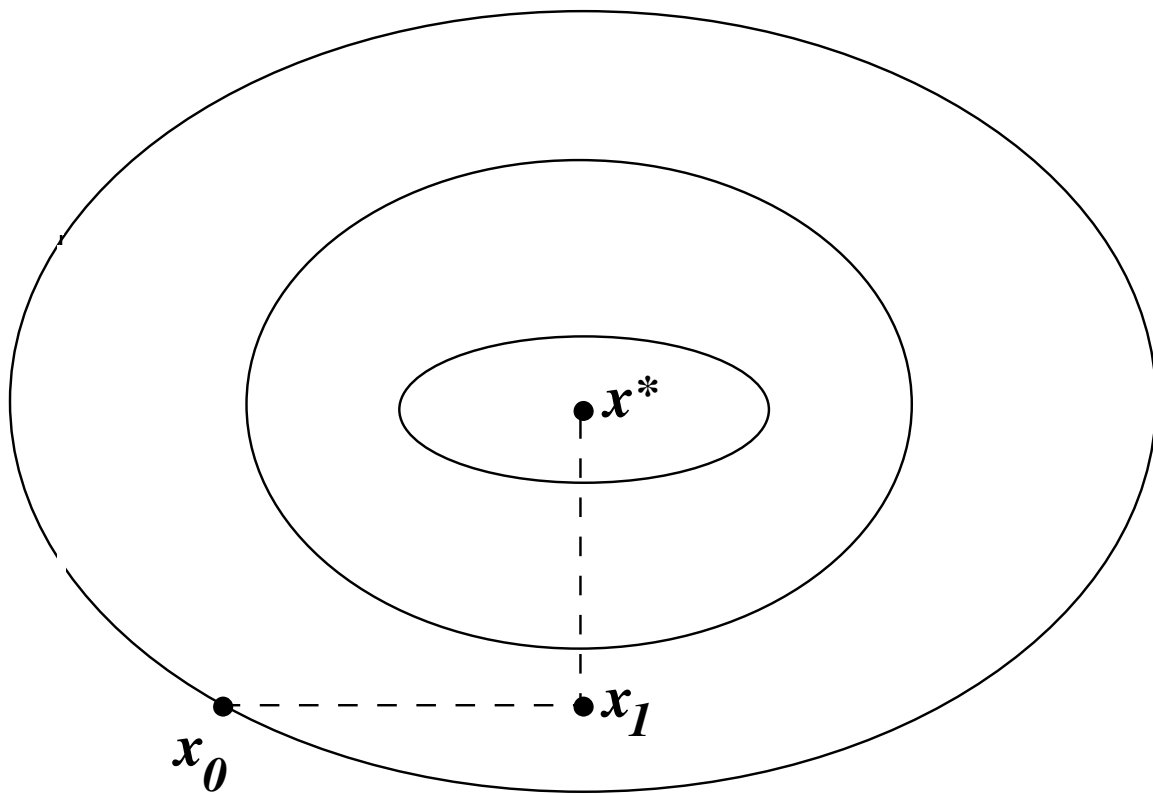
in at most  $n$  steps, and without inverting matrices.

# Conjugate Gradient

Consider the simplest  
diagonal Hessian case first

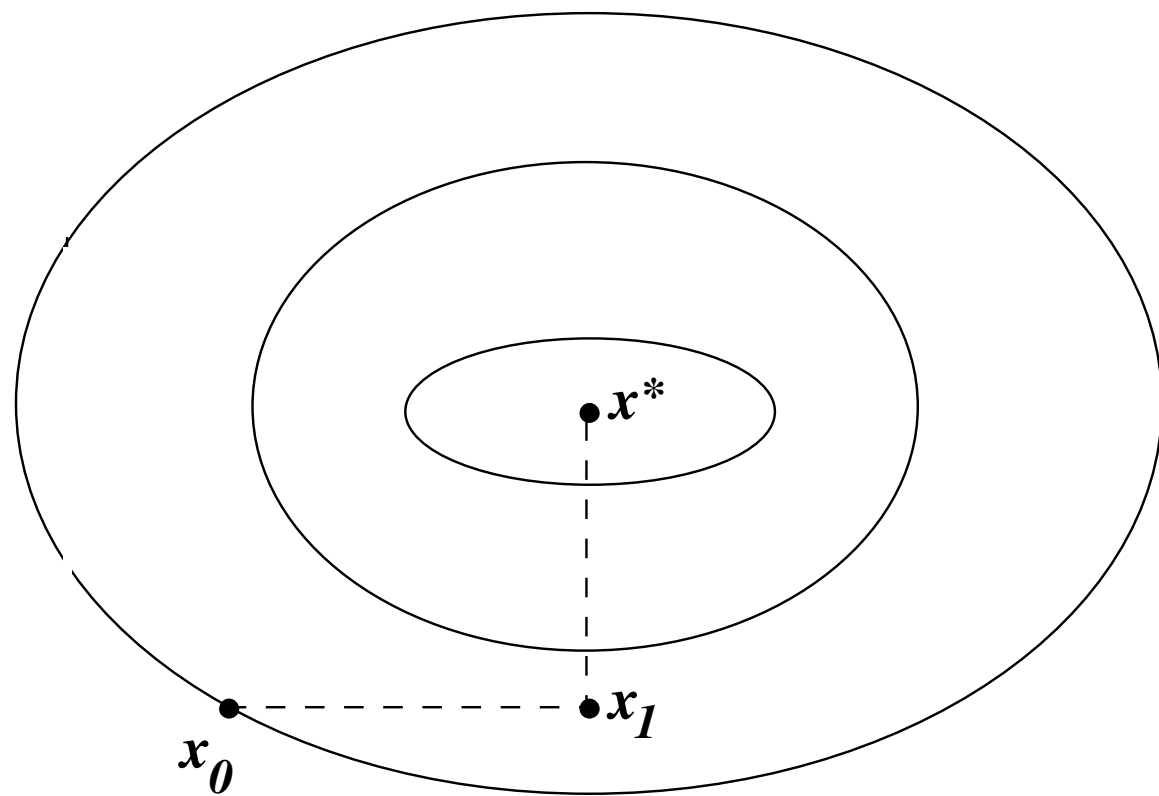
$$f(x) = x_1^2 + 2x_2^2$$

Starting anywhere, say  $(-2, -2)$ ,  
we only need to go in the  
direction of  $x_1$  and  $x_2$  once.



# Conjugate Gradient

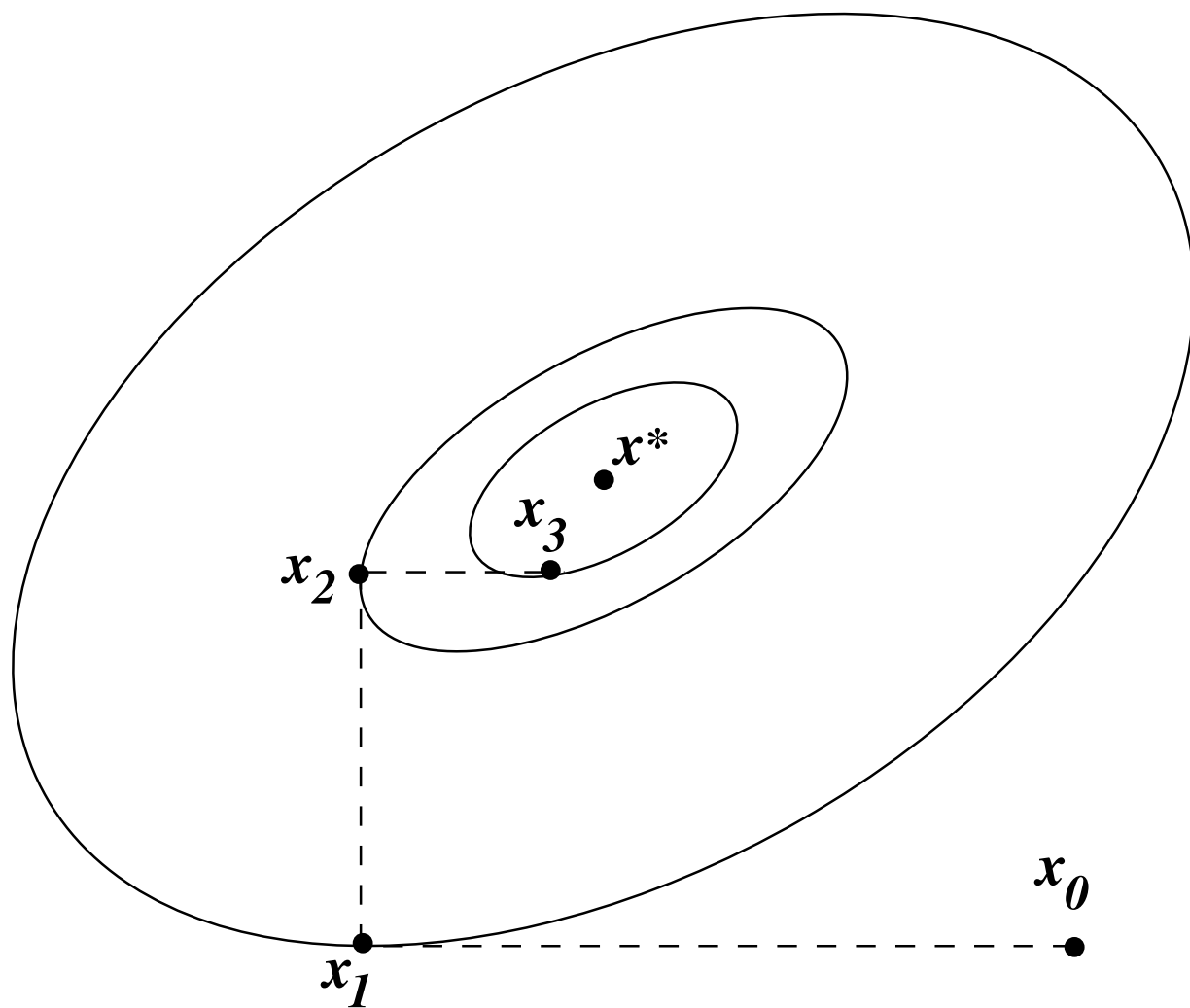
$$f(x) = x_1^2 + 2x_2^2$$



In each direction, we minimize the objective in one big step by choosing  $\alpha_i$  s.t.

$$\frac{\partial f(x + \alpha_i p_i)}{\partial \alpha_i} = 0$$

# Conjugate Gradient



In the general case, the Hessian is not diagonal, and we can no longer just go along the coordinates

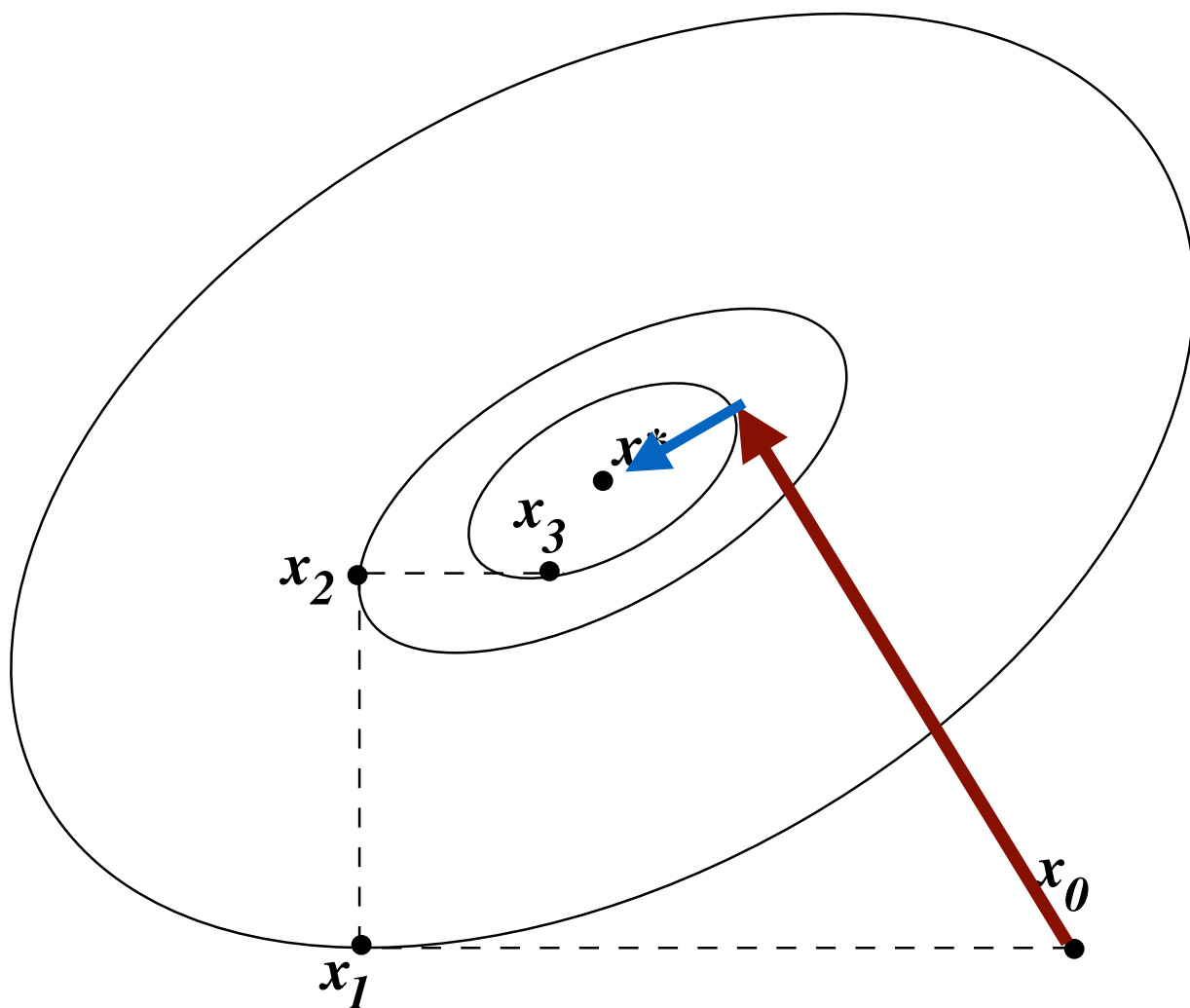
$$f(x) = x^T Q x - b^T x$$

# Conjugate Gradient

In the general case, the Hessian is not diagonal, and we can no longer just go along the coordinates

$$f(x) = x^T Q x - b^T x$$

But we can adjust the directions properly to still quickly move in new coordinates



# Conjugacy

- We say two vectors  $p_1, p_2 \in \mathbb{R}^n$  are conjugate with respect to a positive definite matrix  $Q$  if

$$p_1^T Q p_2 = 0$$

- Any set of  $n$  pairwise nonzero conjugate vectors

$$\{p_0, \dots, p_{n-1}\}$$

forms the conjugate directions over  $\mathbb{R}^n$  w.r.t.  $Q$

# Conjugacy

- Fact: Any set of nonzero pairwise conjugate vectors are linearly independent (and thus spans a subspace).

$$\{p_0, \dots, p_k\}$$

- Proof: Suppose not. Then

$$p_0 = a_1 p_1 + \dots + a_k p_k$$



$$p_0^T Q p_0 = a_1 p_0^T Q p_1 + \dots + a_k p_0^T Q p_k$$

Contradiction.



# Finding Conjugate Directions


- Conjugate directions can be constructed as a combination of the steepest descent direction and the previous direction in each iteration

$$p_k = -r_k + \beta_k p_{k-1}$$

where  $r_k = \nabla_x (\frac{1}{2} x^T Q x - b^T x) |_{x=x_k} = Qx_k - b$

- Multiplying both sides with  $p_{k-1}^T Q$ ,

$$(p_{k-1}^T Q)p_k = - (p_{k-1}^T Q)r_k + (p_{k-1}^T Q)\beta_k p_{k-1}$$

0 

$$\implies \beta_k = p_{k-1}^T Q r_k / p_{k-1}^T Q p_{k-1}$$

# Optimal Steps in Conjugate Directions

- If we have the conjugate directions for  $f(x) = \frac{1}{2}x^T Q x - b^T x$   
 $\{p_0, \dots, p_{n-1}\}$
- Then in each direction, we should move  $\alpha_i$  s.t.

$$\frac{\partial f(x + \alpha_i p_i)}{\partial \alpha_i} = 0$$


Expanding it, we have

$$\frac{\partial f(x + \alpha_i p_i)}{\partial \alpha_i} = \nabla f(x + \alpha_i p_i)^T p_i = \left( (x + \alpha_i p_i)^T Q - b^T \right) p_i = 0$$

# Optimal Steps in Conjugate Directions

- If we have the conjugate directions for  $f(x) = \frac{1}{2}x^T Q x - b^T x$   
 $\{p_0, \dots, p_{n-1}\}$
- Then in each direction, we should move  $\alpha_k$  s.t.

$$\frac{\partial f(x + \alpha_k p_k)}{\partial \alpha_k} = \nabla f(x + \alpha_k p_k)^T p_k = \left( (x + \alpha_k p_k)^T Q - b^T \right) p_k = 0$$

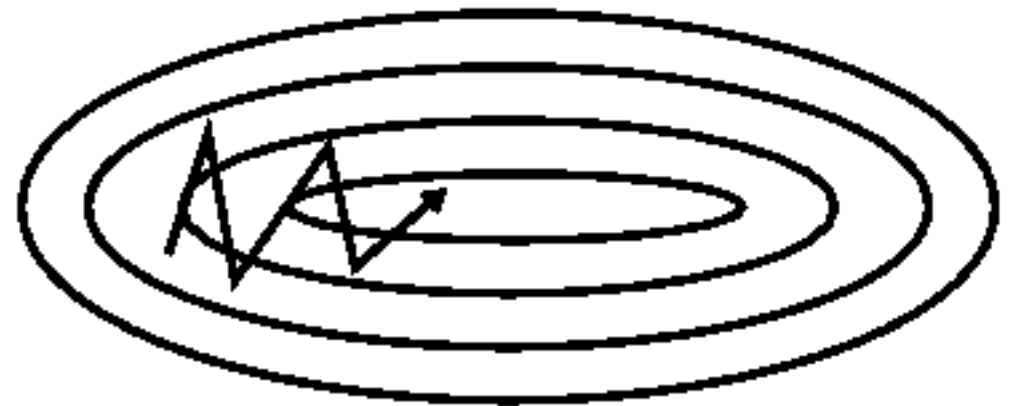

$$\alpha_k = \frac{(b^T - x^T Q) p_k}{p_k^T Q p_k} = \frac{-\nabla f(x)^T p_k}{p_k^T Q p_k}$$

# Conjugate Gradient

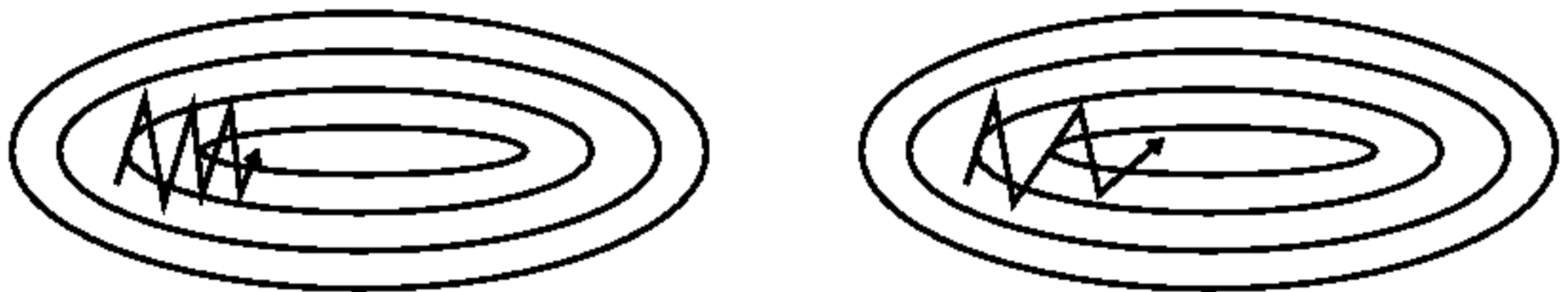
- So the overall CG algorithm is:
  - ▶ Start at arbitrary given  $x_0$ . Set the initial direction to be the steepest  $p_0 = -\nabla f(x_0)$ .
  - ▶ In each direction, first take step  $\alpha_k = \frac{-\nabla f(x_k)p_k}{p_k^T Q p_k}$  then find the next direction  $p_{k+1}$  that is conjugate with  $p_k$  with respect to  $Q$ .
  - ▶ Terminate when  $\nabla f(x_{k+1}) = 0$ .

# Practical Acceleration

- In large-scale learning problems nowadays, we can not afford the computation of the Hessian or frequent evaluation of the function itself.
- A practical trick is the use of “momentum” which is the running average of the gradient



# Momentum

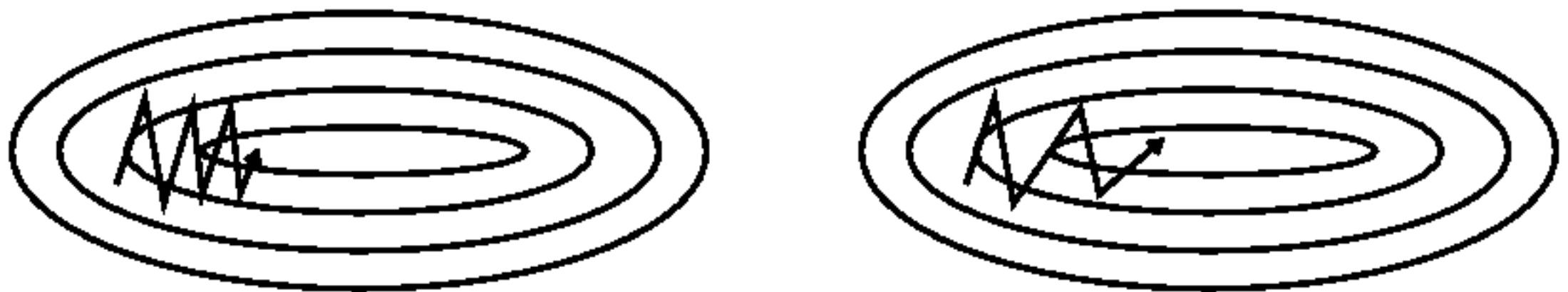


$$v_t = \gamma v_{t-1} + \eta \nabla f(x)$$

$$x = x - v_t \quad \text{typical: } \gamma = 0.9$$

- By averaging, we cancel out the oscillating directions, and speed up in the steady directions.

# Nesterov Accelerated Gradient



$$v_t = \gamma v_{t-1} + \eta \nabla f(x - \gamma v_{t-1})$$

$$x = x - v_t$$

- Looking ahead to roughly where we are going to be, and evaluate the gradient there. More responsive.

# Adagrad (Adaptive Subgradient)

- Automatically tune the learning rate for each dimension of the variables (normalizing)

$$x_{t+1} = x_t - \frac{\eta}{\sqrt{G_t + \varepsilon I}} \nabla f(x_t)$$

where  $G_t = \text{diag}(g_1, \dots, g_n)$  with  $g_i = \sum_{j=1}^t \left( \frac{\partial f(x_j)}{\partial x_{(i)}} \right)^2$



# RMSprop (Root Mean Square Propagation)

- Automatically tune the learning rate for each dimension of the variables (normalizing)

$$G_t = \beta G_{t-1} + (1 - \beta)(\nabla f(x_t))^2$$

$$x_{t+1} = x_t - \frac{\eta}{\sqrt{G_t + \epsilon I}} \nabla f(x_t)$$

which slows down the decay of learning rate

# Adam (Adaptive Moment Estimation)

- Combines Momentum and RMSprop

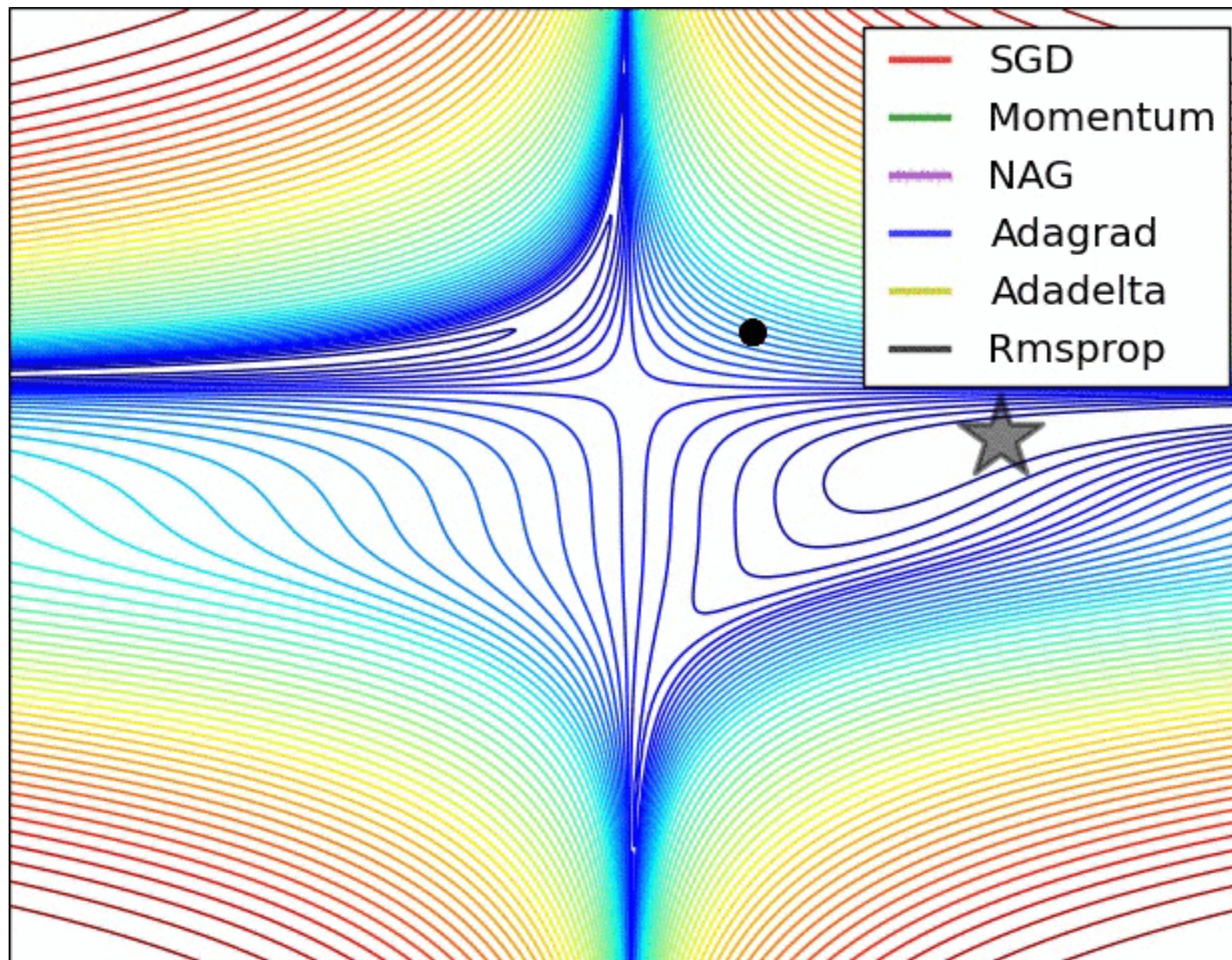
$$v_t = \gamma v_{t-1} + (1 - \gamma) \nabla f(x_t)$$

$$G_t = \beta G_{t-1} + (1 - \beta) (\nabla f(x_t))^2$$

$$x_{t+1} = x_t - \frac{\eta}{\sqrt{\hat{G}_t + \epsilon I}} \hat{v}_t$$

where  $\hat{v}_t = v_t / (1 - \gamma^t)$  and  $\hat{G}_t = G_t / (1 - \beta^t)$

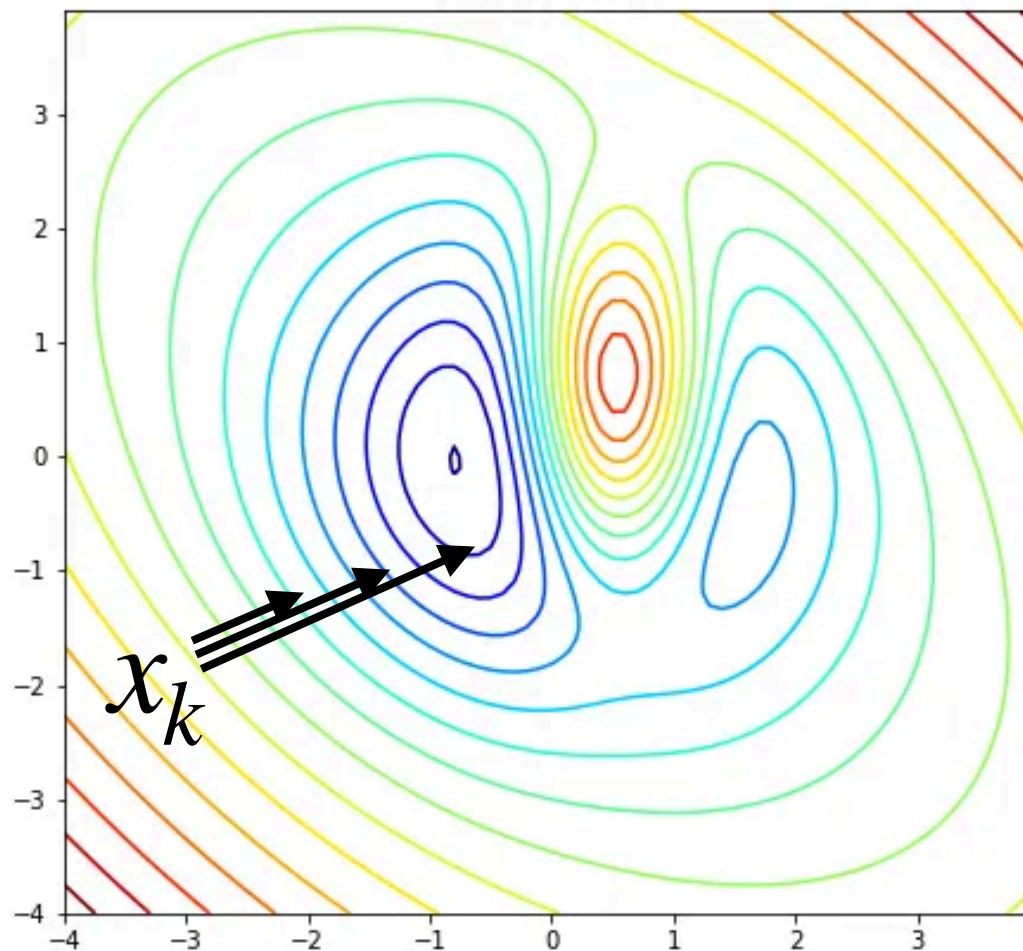
# Practical Development



- Very good simulation to play with: <https://distill.pub/2017/momentum/>



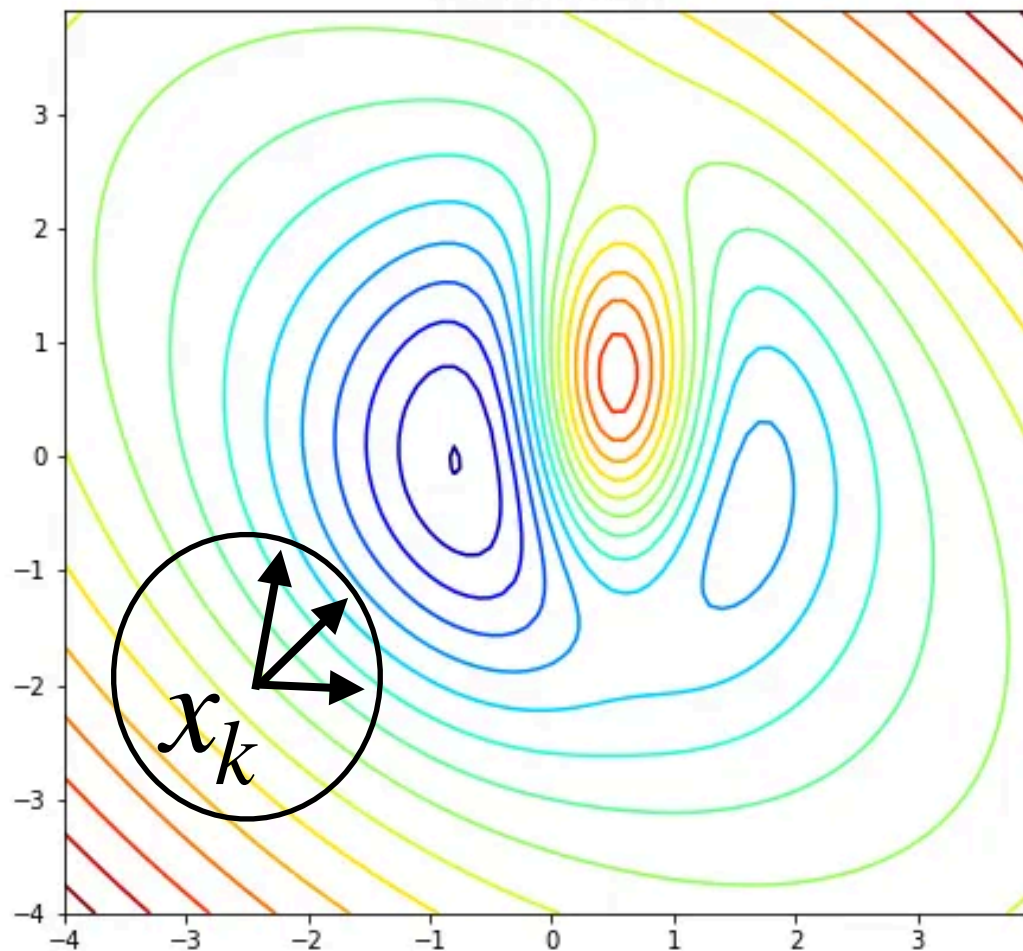
# Line Search vs. Trust Region



Line search methods first choose a direction and then decide how far to go

$$\min_{\alpha > 0} f(x_k + \alpha p_k)$$

# Line Search vs. Trust Region



Trust region methods first construct a model around  $x_k$  and then decide where to go based on the model

$$\min_{p_k \in M(x_k)} f(x_k + p_k)$$