# ASTR8150/PHYS8150
# Optimization

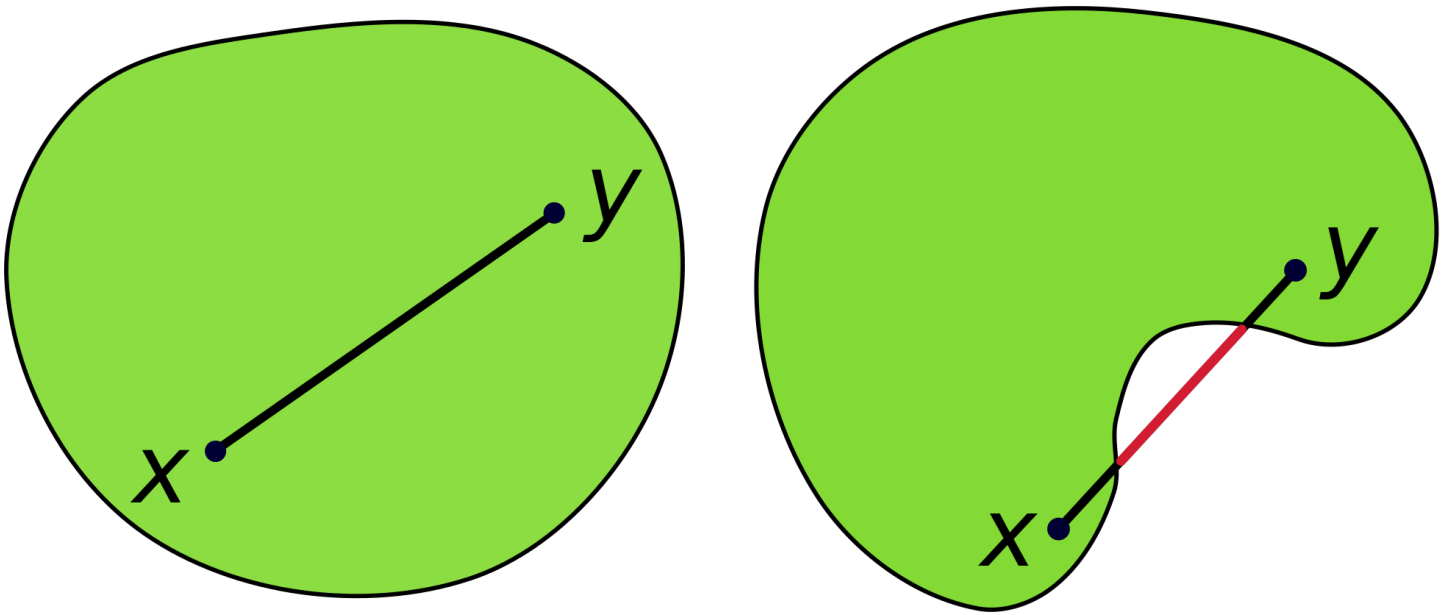
Fabien Baron

Georgia State University

*baron@chara.gsu.edu*

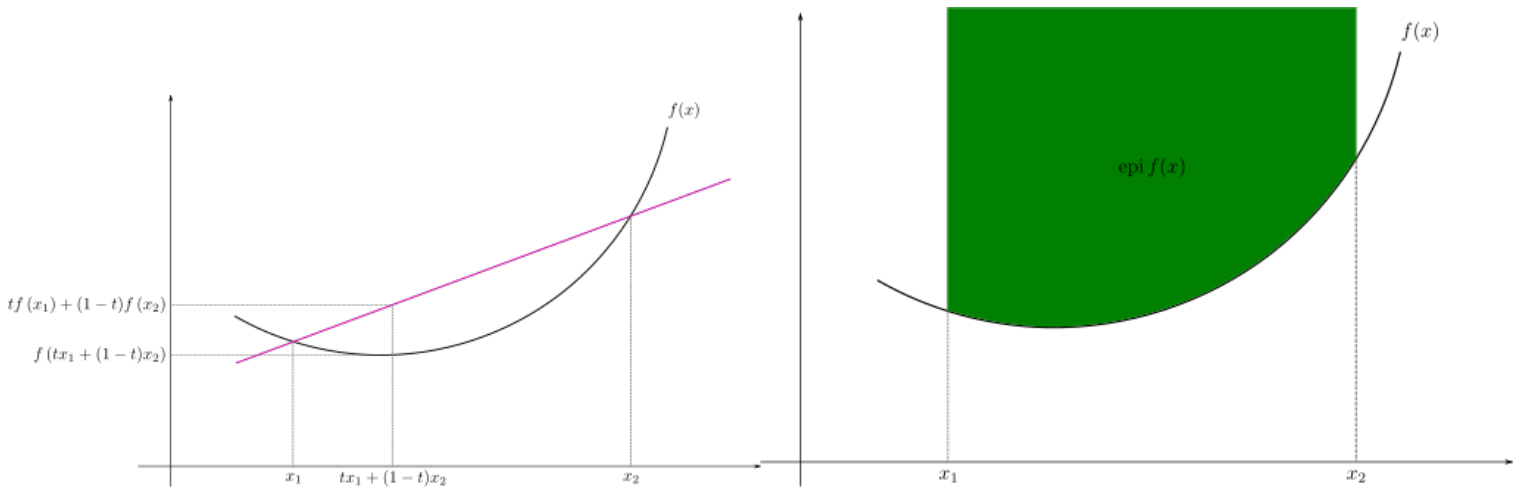
Fall 2017 - Version 1

# Convexity of a set




- In a convex set, for every pair of points within the region, every point on the straight line segment that joins the pair of points is also within the region.
- A set which is hollow or has an indent, for example, a crescent shape, is not convex.

# Convexity of a function



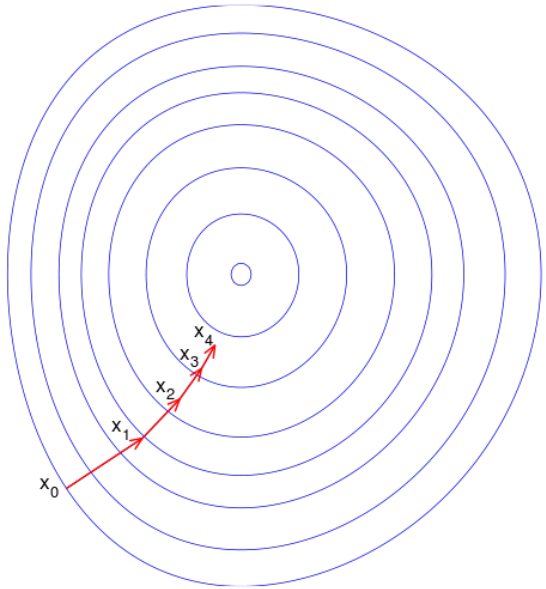

- A real-valued function is called is convex if the set of points on or above the graph of the function (epigraph) is a convex set.
- For a twice differentiable function of a single variable, if the second derivative is always greater than or equal to zero for its entire domain then the function is convex. Examples: $f(x) = x^2$ or $f(x) = e^x$
- Jensen's inequality: if $X$ is a convex set and $f : X \to \mathbb{R}$, $f$ is convex if:

$$\forall x_1, x_2 \in X, \forall t \in [0,1] : \quad f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2).$$

# Smoothness of a function

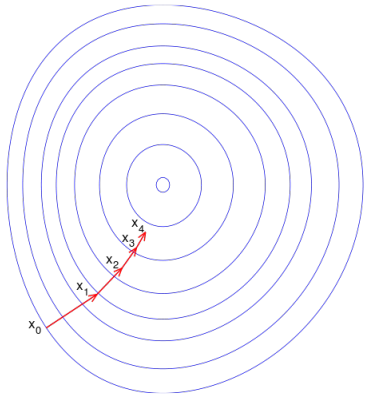
- The smoothness of a function is a property measured by the number of derivatives it has which are continuous. A smooth function is a function that has derivatives of all orders everywhere in its domain.
- The function $f(x) = |x|^k$ is continuous and $k$ times differentiable at all $x$. But at $x = 0$ they are not $(k + 1)$ times differentiable.
- The norms $\ell_1$ and pseudo-norm $\ell_0$ are used in regularization. $\ell_1$ is convex, differentiable but nonsmooth. $\ell_0$ is non-convex and nonsmooth.

# Gradient descent (1)




- Gradient descent is based on the observation that if the multi-variable function $f(\mathbf{x})$ is defined differentiable in a neighborhood of a point $\mathbf{x_0}$, then $F(\mathbf{x})$ decreases "fastest" if one goes from $\mathbf{x_0}$ in the direction of the negative gradient of $f$ at $\mathbf{x_0}$, $-\nabla f(\mathbf{x_0})$.

# Gradient descent (2)




- It follows that, if

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \alpha \nabla f(\mathbf{x}_n) \tag{1}$$

for $\alpha$ small enough, then $f(\mathbf{x_n}) \geq f(\mathbf{x_{n+1}})$. In other words, the term $\alpha \nabla f(\mathbf{x})$ is subtracted from $\mathbf{x}$ because we want to move against the gradient, namely down toward the minimum.

- How can we choose $\alpha$ ?
- The Rosenbrock function $f(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - {x_1}^2)^2$. has a narrow curved valley which contains the minimum. The bottom of the valley is very flat. Because of the curved flat valley the optimization is zig-zagging slowly with small stepsizes towards the minimum.

# Line search

- A line search strategy is one of two basic iterative approaches to find a local minimum $\mathbf{x}^*$ of an objective function $f : \mathbb{R}^n \to \mathbb{R}$. The other approach is trust region.

**Algorithm 1** Line Search

**procedure** LINE SEARCH($f, \mathbf{x}$) ▷ Line search
  $k = 0$, $\mathbf{x}_0$ ▷ Iteration counter + initial parameter guess
  **while** $\|\nabla f(\mathbf{x}_k)\| > \epsilon$ **do** ▷ $\epsilon$ = tolerance
    Descent direction $\mathbf{d}_k$ ▷ $\mathbf{d_k} = -\nabla f(x_k)$ called steepest descent
    $\alpha_k \sim \underset{\alpha \in \mathbb{R}_+}{\text{argmin}}\ f(\mathbf{x}_k + \alpha \mathbf{d}_k)$
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$
    $k = k + 1$
  **end while**
**end procedure**

# Nonlinear conjugate gradient methods

- Let's pose $\mathbf{g_k} = \nabla f(x_k)$
- Conjugate directions deviate from the steepest descent $\mathbf{d_k} = -\mathbf{g_k}$ by attempting moves based on the history of the previous moves
- The descent direction for nonlinear conjugate gradient methods is

$$\mathbf{d_{k+1}} = -\mathbf{g_{k+1}} + \beta_k \mathbf{d_k}, \quad \mathbf{d_0} = -\mathbf{g_0} \tag{2}$$

- The variation of the gradient is measured by $\mathbf{y_k} = \mathbf{g_{k+1}} - \mathbf{g_k}$
- The Conjugate Gradient update parameter $\beta_k$ can be updated with different formulas

# Nonlinear conjugate gradient methods

$$\beta_k^{HS} = \frac{\mathbf{g}_{k+1}^\mathsf{T}\mathbf{y}_k}{\mathbf{d}_k^\mathsf{T}\mathbf{y}_k}$$

(1952) in the original (linear) CG paper of Hestenes and Stiefel [59]

$$\beta_k^{FR} = \frac{\|\mathbf{g}_{k+1}\|^2}{\|\mathbf{g}_k\|^2}$$

(1964) first nonlinear CG method, proposed by Fletcher and Reeves [45]

$$\beta_k^{D} = \frac{\mathbf{g}_{k+1}^\mathsf{T}\nabla^2 f(\mathbf{x}_k)\mathbf{d}_k}{\mathbf{d}_k^\mathsf{T}\nabla^2 f(\mathbf{x}_k)\mathbf{d}_k}$$

(1967) proposed by Daniel [39], requires evaluation of the Hessian $\nabla^2 f(\mathbf{x})$

$$\beta_k^{PRP} = \frac{\mathbf{g}_{k+1}^\mathsf{T}\mathbf{y}_k}{\|\mathbf{g}_k\|^2}$$

(1969) proposed by Polak and Ribière [84] and by Polyak [85]

$$\beta_k^{CD} = \frac{\|\mathbf{g}_{k+1}\|^2}{-\mathbf{d}_k^\mathsf{T}\mathbf{g}_k}$$

(1987) proposed by Fletcher [44], CD stands for "Conjugate Descent"

$$\beta_k^{LS} = \frac{\mathbf{g}_{k+1}^\mathsf{T}\mathbf{y}_k}{-\mathbf{d}_k^\mathsf{T}\mathbf{g}_k}$$

(1991) proposed by Liu and Storey [67]

$$\beta_k^{DY} = \frac{\|\mathbf{g}_{k+1}\|^2}{\mathbf{d}_k^\mathsf{T}\mathbf{y}_k}$$

(1999) proposed by Dai and Yuan [27]

$$\beta_k^{N} = \left(\mathbf{y}_k - 2\mathbf{d}_k\frac{\|\mathbf{y}_k\|^2}{\mathbf{d}_k^\mathsf{T}\mathbf{y}_k}\right)^\mathsf{T}\frac{\mathbf{g}_{k+1}}{\mathbf{d}_k^\mathsf{T}\mathbf{y}_k}$$

(2005) proposed by Hager and Zhang [53]

# Newton optimization method

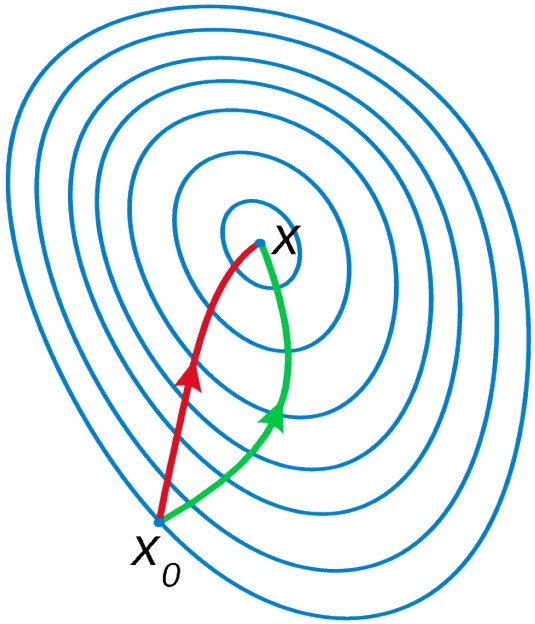



Figure: A comparison of gradient descent (green) and Newton's method (red) for minimizing a function (with small step sizes). Newton's method uses curvature information to take a more direct route.

- Hessian is used to exploit the curvature information

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \alpha[\mathbf{H}f(\mathbf{x}_n)]^{-1}\nabla f(\mathbf{x}_n) \tag{3}$$

- $\alpha \in (0, 1)$, with $\alpha = 1$ the exact form.

# Newton-Raphson root-finding and eccentric anomaly

- Newton optimization method and Newton-Raphson's root finding methods are based on similar principles
- Newton-Raphson: $x_{n+1} = x_n - f(x_n)/f'(x_n)$
- Example: the mean anomaly is proportional to time it is an easily measured quantity for an orbiting body. Given the mean anomaly $M$, find the eccentric anomaly $E$ and the orbital eccentricity $e$ with Kepler's Equation:

$$M = E - e \sin E \tag{4}$$

# Better than Newton: semi-Newton methods

- Also known as variable metric methods, they avoid computing the Hessian then its inverse.
- The Broyden-Fletcher-Goldfarb-Shanno (BFGS) or Davidon-Fletcher-Powell (DFP) algorithms build iteratively approximations of $[\mathbf{H}f(\mathbf{x}_n)]^{-1}$.
- The most successfull is Limited-memory BFGS (L-BFGS) that approximates $[\mathbf{H}f(\mathbf{x}_n)]^{-1}\nabla f(\mathbf{x}_n)$ directly and thus can work on large scale problems (millions of variables).
- There are variants that attempt to deal with non-smooth functions (subgradient and bundle method)
- There are variants that deal with constrained miminization (i.e. bounds on variables or linearly tied variables) such as L-BFGS-B. Further refinements led to the VMLM algorithm in OptimPack.

# Trust-region method and Levenberg-Marquardt

- Consider the quadratic approximation of function f around $x_0$:

$$q(\epsilon) \simeq f(x_0) + \nabla f(x_0)\epsilon + \frac{1}{2}\epsilon^T \nabla^2 f(x_0)\epsilon \tag{5}$$

- $q(\epsilon)$ has a close-form mimimum.
- $q(\epsilon)$ remains a good approximation within a given radius, $||e||_2 < r^2$ defines the **trust region** radius $r$.
- The quadratic approximation predicts a certain reduction in the cost function, $\Delta f_{\text{pred}}$, which is compared to the true reduction $\Delta f_{\text{actual}} = f(x) - f(x+\epsilon)$. By looking at the ratio $\Delta f_{\text{pred}}/\Delta f_{\text{actual}}$ we can estimate the trust-region size at each iteration, jump to the closed-form minimum within the trust region, and iterate.
- The Levenberg-Marquardt algorithm (first published in 1944 by Kenneth Levenberg, rediscovered in 1963 by Donald Marquardt) uses the trust-region approach with conjugate-gradients and Gauss-Newton (Newton optimized for non-linear $\chi^2$). Like conjugate gradient and Newton, these local optimization.

# Derivative-free optimization methods

- Among the most popular local optimizer is NelderMead method (aka downhill simplex method or amoeba method), which moves points of a polytope of $n+1$ vertices in $n$-parameter dimensions via reflection, contraction, expansion steps
- Most MCMC optimization methods.
- NLopt library provides mostly derivative-free algorithms, some of them for global optimization.