Lecture 2: Review, Contour Maps, Various Forms of Generalized Gradient Methods, and Line Search Methods

Prof. Krishna R. Pattipati
Dept. of Electrical and Computer Engineering
University of Connecticut
Contact: krishna@engr.uconn.edu (860) 486-2890

ECE 6437
Computational Methods for Optimization

Fall 2009
September 8, 2009

Outline of Lecture 2

- Review of Lecture 1
- Gradients and Contour Maps
- Algorithms for Unconstrained Minimization … Answers to the Dynamic Question
- Generalized Gradient Methods
- Step Size Rules (or) Line Search Methods

Review of Lecture 1

- Necessary and sufficient conditions for a local minimum

  Necessary conditions

  $$\nabla \underline{f}(\underline{x}^*) = \underline{0}$$

  $$\nabla^2 f(\underline{x}^*) \geq 0$$

  Sufficient conditions

  $$\nabla \underline{f}(\underline{x}^*) = \underline{0}$$

  $$\nabla^2 f(\underline{x}^*) > 0$$

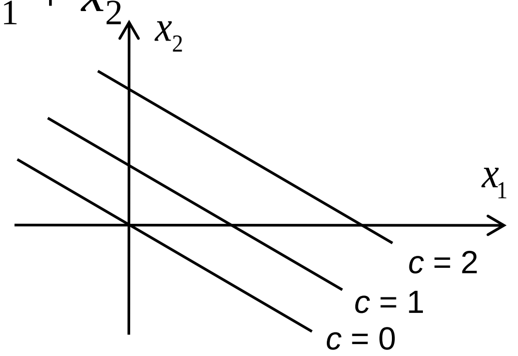
  For general $f(\underline{x})$ local minimum $\not\Rightarrow$ global minimum

  For convex $f(\underline{x})$ local minimum $\Leftrightarrow$ global minimum

- Gradient and contour maps

  Contour or equivalent surface: $f(\underline{x}) = c \;\Rightarrow\; \{x \in \Omega : f(\underline{x}) = c\}$

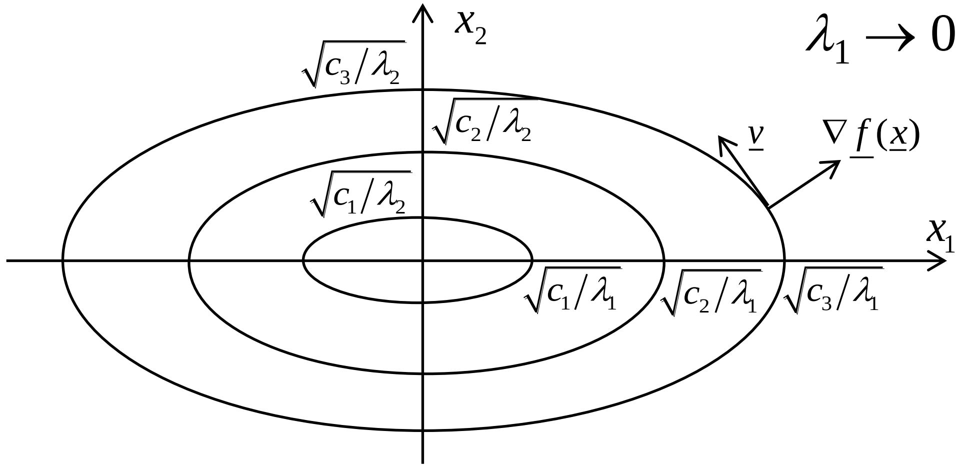
- Example 1: $f(x_1, x_2) = x_1 + x_2$



Gradients and Contour Maps

- Example 2: $f(x_1, x_2) = \lambda_1 x_1^2 + \lambda_2 x_2^2 \;\; ; \;\; \lambda_2 > \lambda_1 > 0$

$\lambda_1 \to 0 \;\Rightarrow$ long elongated ellipse



The contour curve can be parameterized by $x_1(t), x_2(t)$

$$f(x_1(t),\ x_2(t)) = c$$

$$\Rightarrow \quad \frac{\partial f}{\partial x_1}\cdot\frac{dx_1}{dt} + \frac{\partial f}{\partial x_2}\cdot\frac{dx_2}{dt} = \frac{df}{dt} = 0$$

GRADIENTS ARE ORTHOGONAL TO CONTOUR CURVES

$$\begin{bmatrix} \dfrac{\partial f}{\partial x_1} & \dfrac{\partial f}{\partial x_2} \end{bmatrix} \begin{bmatrix} \dfrac{dx_1}{dt} \\ \dfrac{dx_2}{dt} \end{bmatrix} = 0 \;\Rightarrow\; \nabla \underline{f}^T(\underline{x})\ \underline{v} = 0$$

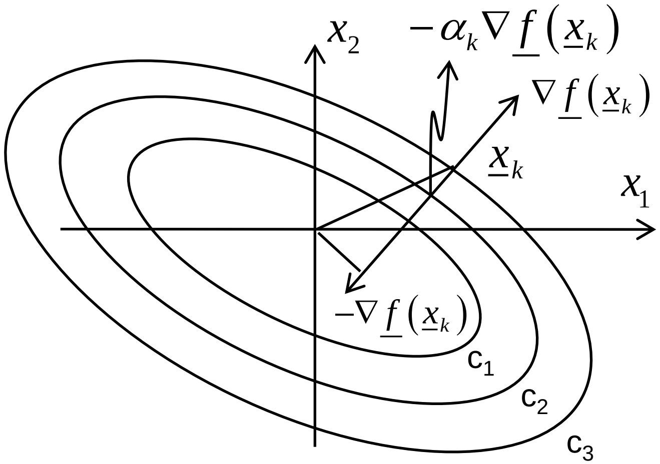
Negative Gradient as Descent Direction

- Want $\underline{x}_0 \rightarrow \underline{x}_1 \rightarrow \underline{x}_2 \rightarrow \ldots \rightarrow \underline{x}_k \rightarrow \underline{x}_{k+1} \rightarrow \ldots \rightarrow \underline{x}^* \ni f(\underline{x}_0) \geq f(\underline{x}_1) \geq f(\underline{x}_2) \geq \ldots \geq f(\underline{x}^*)$

  $\Rightarrow$ f is decreased at each iteration (or) we move from one contour to the next such that $c_k \geq c_{k+1}$. **DESCENT ALGORITHMS**

- **Q:** How do we move from $\underline{x}_k$ to $\underline{x}_{k+1} \ni f(\underline{x}_{k+1}) \leq f(\underline{x}_k)$?

  1. Recall that $\nabla \underline{f}(\underline{x}_k)$ is the direction of increase in f at $\underline{x} = \underline{x}_k$ then $-\nabla \underline{f}(\underline{x}_k)$ is the direction of (local) decrease in f. So, one way to move from $\underline{x}_k$ to $\underline{x}_{k+1}$ is via: $\underline{x}_{k+1} = \underline{x}_k - \alpha_k \nabla \underline{f}(\underline{x}_k), \quad \alpha_k \geq 0$



Steepest descent, Gradient or Cauchy's method

From Taylor series expansion

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

$$= f(\underline{x}_k) - \alpha_k \nabla \underline{f}^T(\underline{x}_k) \nabla \underline{f}(\underline{x}_k) + O(\alpha_k^2)$$

$\Rightarrow$ For sufficiently small α_k: $f(\underline{x}_{k+1}) < f(\underline{x}_k)$

More General Descent Directions

2. What are the general directions we can take to go from $\underline{x}_k$ to $\underline{x}_{k+1}$?

$$\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{d}_k$$

$$\underline{d}_k = -\nabla \underline{f}(\underline{x}_k) \quad \Rightarrow \quad \text{Gradient method}$$

What are the restrictions on $\underline{d}_k$ to ensure $f(\underline{x}_{k+1}) < f(\underline{x}_k)$?

– By def^n: $f(\underline{x}_{k+1}) = f(\underline{x}_k + \alpha \underline{d}_k) = f(\underline{x}_k) + \alpha \underbrace{\nabla \underline{f}^T(\underline{x}_k)\underline{d}_k}_{\text{Directional derivative}} + O(\alpha^2)$

$\nabla \underline{f}^T(\underline{x}_k)\underline{d}_k$	=	Directional derivative of f at $\underline{x}_k$ in the direction $\underline{d}_k$
	=	Rate of change of f in the direction $\underline{d}_k$ with respect to α
	=	Inner product of the gradient at $\underline{x}_k$ and the selected direction $\underline{d}_k$

– For sufficiently small α, we can guarantee $f(\underline{x}_{k+1}) < f(\underline{x}_k)$ if $\boxed{\nabla \underline{f}^T(\underline{x}_k)\underline{d}_k < 0}$

– If $\nabla \underline{f}^T(\underline{x}_k)\underline{d}_k < 0 \Rightarrow \underline{d}_k$ is the descent direction since it ensures a reduction in the function value

– Recall that $\nabla \underline{f}^T(\underline{x}_k)\underline{d}_k = \left\|\nabla \underline{f}(\underline{x}_k)\right\|_2 \left\|\underline{d}_k\right\|_2 \cos\theta$

$$\nabla \underline{f}^T(\underline{x}_k)\underline{d}_k < 0 \quad \Rightarrow \quad \theta \in (90^o,\ 180^o) \text{ or } (-180^o,\ -90^o)$$

A General Form for Descent Direction

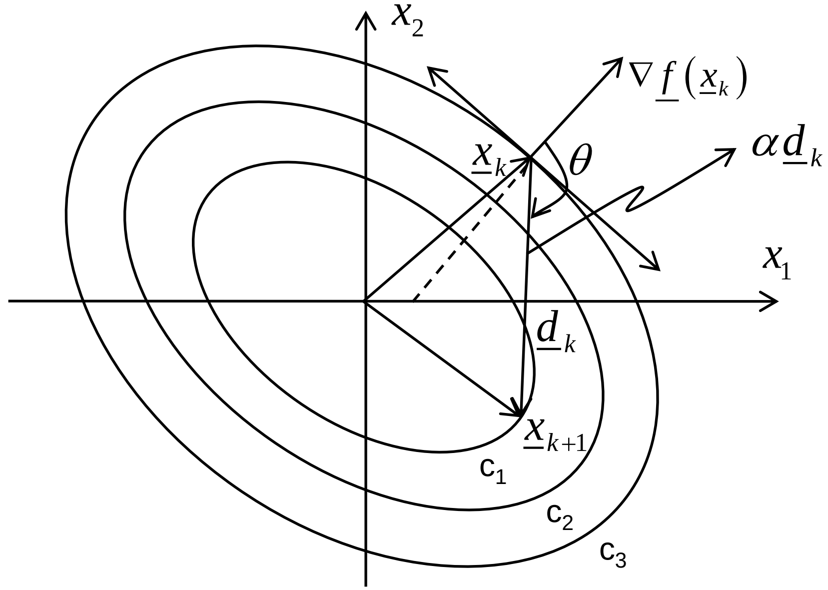
– A general form of $\underline{d}_k$:

$$\underline{d}_k = -H_k \nabla \underline{f}(\underline{x}_k), \quad H_k > 0$$

$$\Rightarrow \quad \nabla \underline{f}^T(\underline{x}_k)\underline{d}_k = -\nabla \underline{f}^T(\underline{x}_k) H_k \nabla \underline{f}(\underline{x}_k) < 0, \quad \forall \; \nabla \underline{f}(\underline{x}_k) \neq 0$$

– If $\|\underline{d}_k\| < \infty$, $\underline{d}_k$ is termed "uniformly gradient related"

- Note: $H_k = I \Rightarrow$ steepest descent



Generalized Gradient Methods (GGM)

$$\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{d}_k = \underline{x}_k - \alpha_k H_k \nabla \underline{f}\left(\underline{x}_k\right), \quad k = 0,\ 1,\ 2,\ \ldots$$

- Key questions:

1. How to pick $\underline{d}_k$ (or equivalently H_k)
2. How to pick α_k to get a "good sized" reduction in the function value

   2.1 Pick α_k to get any decrease in function value

   $f\left(\underline{x}_{k+1}\right) < f\left(\underline{x}_k\right)$ ... won't work !!

   2.2 Pick α_k to get a specified decrease in function value

   Armijo and Goldstein step size rules

   2.3 Pick α_k to get maximum decrease in function value

   $$f\left(\underline{x}_k + \alpha_k \underline{d}_k\right) = \min_{\alpha \geq 0} f\left(\underline{x}_k + \alpha \underline{d}_k\right)$$

Ways to Pick Descent Directions - 1

- How to pick $\underline{d}_k$ (or H_k) . . . Determines the algorithm
  1. Steepest descent (gradient, Cauchy) method

     $$H_k = I \;\Rightarrow\; \underline{d}_k = -\nabla \underline{f}(\underline{x}_k),\;\; k = 0, 1, 2, \ldots$$

  2. Diagonally scaled steepest descent method

     $$H_k = \mathrm{Diag}\left(h_k^1,\, h_k^2,\, \ldots,\, h_k^n\right),\;\; k = 0, 1, 2, \ldots$$

     $$h_k^i > 0,\;\; \text{quite often } h_k^i = \left[\frac{\partial^2 f(\underline{x}_k)}{\partial x_i^2}\right]^{-1} \longrightarrow \boxed{\text{Diagonal Newton's Method}}$$

  3. Newton's method

     $$H_k = \left[\nabla^2 f(\underline{x}_k)\right]^{-1},\;\; k = 0, 1, 2, \ldots \text{ if } \nabla^2 f(\underline{x}_k) > 0$$

     Generally, implemented as $H_k = \left(L_k^T\right)^{-1} \tilde{D}_k^{-1} L_k^{-1}$ where $\nabla^2 \underline{f}(\underline{x}_k) = L_k \tilde{D}_k L_k^T$

     – Don't actually compute the inverse: $\underline{d}_k = -\left[\nabla^2 f(\underline{x}_k)\right]^{-1} \nabla \underline{f}(\underline{x}_k)$

     Instead solve $\left[\nabla^2 f(\underline{x}_k)\right] \underline{d}_k = -\nabla \underline{f}(\underline{x}_k)$

     (or) $L_k \tilde{D}_k L_k^T\, \underline{d}_k = -\nabla \underline{f}(\underline{x}_k)$ →

     - solve $L_k \underline{y}_k = -\nabla \underline{f}(\underline{x}_k)$
     - $z_k^i = y_k^i / \tilde{d}_k^i,\;\; 1 \le i \le n$
     - $L_k^T \underline{d}_k = \underline{z}_k$

$\boxed{\approx O(n^2) \text{ operations once } L_k \text{ and } \tilde{D}_k \text{ are available}}$

Ways to Pick Descent Directions - 2

– Solve $f(\underline{x}_k) = \frac{1}{2}\underline{x}^T Q \underline{x} + \underline{b}^T \underline{x} + \underline{c}$ (quadratic form) in one iteration.

$$\boxed{\begin{aligned}\underline{x}_1 &= \underline{x}_0 - Q^{-1}(Q\underline{x}_0 + \underline{b}) \\ &= -Q^{-1}\underline{b} = \underline{x}^*\end{aligned}}$$

4. Modified Newton's method
   – Modify H_k as:

$$H_k = \begin{cases} \left[\nabla^2 f(\underline{x}_k)\right]^{-1} & \text{if PD} \\ \left[\nabla^2 f(\underline{x}_k) + \mu_k I\right]^{-1} & \text{if not PD} \end{cases}$$

This can be accomplished as part of LDL^T factorization

This modification is due to Levenberg and Marquardt (1944, 1963)

– $H_k = \left[\nabla^2 f(\underline{x}_0)\right]^{-1},\ k = 0, 1, 2, \ldots$

Use Hessian at the starting point $\underline{x}_0$ only

Need to compute LDL^T decomposition once !! $\Rightarrow O\left(\frac{n^3}{6}\right)$ computation

– $H_{kp+i} = \left[\nabla^2 f(\underline{x}_{kp})\right]^{-1};\ i = 0, 1, 2, \ldots, p-1;\ k = 0, 1, 2, \ldots$

if $\nabla^2 f(\underline{x}_{kp})$ is PD

Ways to Pick Descent Directions - 3

5. Discretized Newton's method

$$\frac{\partial^2 f}{\partial x_i^2} = \frac{\dfrac{\partial f(\underline{x}_k)}{\partial x_i} - \dfrac{\partial f(\underline{x}_k - \underline{e}_i\delta)}{\partial x_i}}{\delta}; \quad \frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\dfrac{\partial f(\underline{x}_k)}{\partial x_i} - \dfrac{\partial f(\underline{x}_k - \underline{e}_j\delta)}{\partial x_i}}{\delta}$$

Use these to approximate $H_k = \left[\left\{\nabla^2 f\left(\underline{x}_k\right)\right\}_d\right]^{-1}$

6. Gauss-Newton method (specifically for nonlinear least squares problem. Levenberg -Marquardt was developed in this context)

$$\underline{g} = \left(g_1\ g_2\ \ldots\ g_m\right)^T$$

$$\min_{\underline{x}} f(\underline{x}) = \frac{1}{2}\underline{g}^T(\underline{x})\underline{g}(\underline{x}) = \frac{1}{2}\sum_{i=1}^{m} g_i^2(\underline{x})$$

$$H_k = \begin{cases} \left[\nabla \underline{\underline{g}}(\underline{x}_k)\nabla \underline{\underline{g}}^T(\underline{x}_k)\right]^{-1} & \text{if invertible} \\ \left[\nabla \underline{\underline{g}}(\underline{x}_k)\nabla \underline{\underline{g}}^T(\underline{x}_k) + \mu I\right]^{-1} & \text{if not} \end{cases}$$

Ways to Pick Descent Directions - 4

- Note 1:

$$\nabla \underline{f}(\underline{x}_k) = \nabla \underline{\underline{g}}(\underline{x}_k)\underline{g}(\underline{x}_k)$$

$$\nabla \underline{\underline{g}}(\underline{x}_k) = \left[\nabla \underline{g}_1(\underline{x}_k)\ \nabla \underline{g}_2(\underline{x}_k)\ \ldots\ \nabla \underline{g}_m(\underline{x}_k)\right] \quad n \text{ by } m \text{ matrix}$$

$$\nabla \underline{\underline{g}}^T \sim \text{Jacobian}$$

For this problem, Gauss-Newton method takes the form:

$$\underline{x}_{k+1} = \underline{x}_k - \alpha_k \left[\nabla \underline{\underline{g}}(\underline{x}_k)\nabla \underline{\underline{g}}^T(\underline{x}_k)\right]^{-1} \nabla \underline{\underline{g}}(\underline{x}_k)\underline{g}(\underline{x}_k)$$

- Note 2:
  1) Levenberg-Marquardt iteration first appeared in this form
  2) This iteration also occurs in maximum likelihood (ML) identification of linear dynamic systems in a slightly complex form
  3) Incremental Gradient (used in Neural network training)
  4) Extended Kalman filter is basically an incremental version of Gauss-Newton

Ways to Pick Descent Directions - 5

7. Conjugate gradient method
   – Seeks to generate $\underline{d}_k$ directly without having to form H_k

$$\underline{d}_k = -\nabla \underline{f}\left(\underline{x}_k\right) + \beta_k \underline{d}_{k-1}$$

$$\beta_k = \begin{cases} \dfrac{\nabla \underline{f}^T\left(\underline{x}_k\right)\nabla \underline{f}\left(\underline{x}_k\right)}{\nabla \underline{f}^T\left(\underline{x}_{k-1}\right)\nabla \underline{f}\left(\underline{x}_{k-1}\right)} & \text{Fletcher-Reeves (FR)method} \\ \dfrac{\nabla \underline{f}^T\left(\underline{x}_k\right)\left[\nabla \underline{f}\left(\underline{x}_k\right) - \nabla \underline{f}\left(\underline{x}_{k-1}\right)\right]}{\nabla \underline{f}^T\left(\underline{x}_{k-1}\right)\nabla \underline{f}\left(\underline{x}_{k-1}\right)} & \text{Polar-Ribiere-Poljak (PRP) method} \\ \dfrac{\nabla \underline{f}^T\left(\underline{x}_k\right)\left[\nabla \underline{f}\left(\underline{x}_k\right) - \nabla \underline{f}\left(\underline{x}_{k-1}\right)\right]}{\left[\nabla \underline{f}\left(\underline{x}_k\right) - \nabla \underline{f}\left(\underline{x}_{k-1}\right)\right]^T \underline{d}_{k-1}} & \text{Sorensen-Wolfe (SW) method} \end{cases}$$

   – Solves $f\left(\underline{x}\right) = \frac{1}{2}\underline{x}^T Q \underline{x} + \underline{b}^T \underline{x} + \underline{c}$ in $\underline{n}$ iterations (quadratic termination property)

Ways to Pick Descent Directions - 6

8. Quasi-Newton (or) Variable Metric Methods
   – Since Newton's method minimizes quadratic functions in <u>one</u> iteration, how about approximating inverse of Hessian or Hessian as a function of $\underline{x}_k$ and $\nabla \underline{f}(\underline{x}_k)$ etc.

INVERSE: APPROX.

$$H_{k+1} = H_k + \frac{\underline{p}_k \underline{p}_k^T}{\underline{p}_k^T \underline{q}_k} - \frac{H_k \underline{q}_k \underline{q}_k^T H_k}{\underline{q}_k^T H_k \underline{q}_k}$$

Davidon-Fletcher-Powell (DFP) update

HESSIAN: APPROX.

$$B_{k+1} = B_k + \frac{\underline{q}_k \underline{q}_k^T}{\underline{p}_k^T \underline{q}_k} - \frac{B_k \underline{p}_k \underline{p}_k^T B_k}{\underline{p}_k^T B_k \underline{p}_k}$$

Broyden-Fletcher-Goldfarb-Shanno (BFGS) update

$$\underline{p}_k = \underline{x}_{k+1} - \underline{x}_k = \alpha_k \underline{d}_k$$

$$\underline{q}_k = \nabla \underline{f}(\underline{x}_{k+1}) - \nabla \underline{f}(\underline{x}_k) = \underline{g}_{k+1} - \underline{g}_k$$

   – Has quadratic termination property (converges in n iterations for quadratic functions
   – Note: don't need second derivative information

What do we want to do with these?

- Key questions:

  1) Do the methods converge?
  2) If so, do they converge to a local minimum or a stationary point (i.e., maximum, saddle point, minimum)?
  3) What is the order (speed) of convergence ?
  4) How does the choice of α_k affect convergence ?

  $\Rightarrow$ We will focus on the problem of selecting α_k next.

Step Size Rules

- Step size rules (or) Line search methods:
  - Pick α_k to
    - Guarantee a reduction in function value $f(\underline{x}_{k+1}) < f(\underline{x}_k)$
    - Guarantee at least a specified reduction in the function value
    - Minimize $f(\underline{x}_k + \alpha\, \underline{d}_k)$ with respect to α such that $\alpha > 0$

Increasing complexity

  - Know that if $\nabla \underline{f}(\underline{x}_k) \neq 0$. Then $\exists$ a sufficiently small $\alpha \ni$ $f(\underline{x}_{k+1}) < f(\underline{x}_k)$ since $\nabla \underline{f}^T(\underline{x}_k)\underline{d}_k \neq 0$. We will show that a mere guarantee of reduction in f does not result in a reliable algorithm in the sense that the sequence $\{\underline{x}_k\}$ may converge to a nonstationary point !!
    - Consider steepest descent iteration

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

and the following algorithm for finding α_k

Naïve Methods May Not Work

Given $s > 0,\ \beta \in (0,\ 1),$ find $f\left(\underline{x}_k - s\nabla \underline{f}(\underline{x}_k)\right)$

Do while $f\left(\underline{x}_k - s\nabla \underline{f}(\underline{x}_k)\right) \geq f(\underline{x}_k)$

$$s \leftarrow s\beta$$

Evaluate $f\left(\underline{x}_k - s\nabla \underline{f}(\underline{x}_k)\right)$

end Do

$\alpha_k = s$

$$s = \left| \frac{f(\underline{x}_k).\ \rho.\ \gamma}{\nabla \underline{f}^T(\underline{x}_k).\underline{d}_k} \right|$$

$\rho \approx [0.1,\ 0.3];\ \gamma \approx [2,\ 10]$

(or) see three point pattern method later

- PROBLEM: Can get stuck at a non-stationary point.
- Note: $g(0) = f(\underline{x}_k);$

$$g(\alpha) = f(\underline{x}_k + \alpha\underline{d}_k) = g(0) + \alpha g'(0) + \ldots$$

$$g'(\alpha) = \nabla \underline{f}^T(\underline{x}_k + \alpha\underline{d}_k)\underline{d}_k;\quad g'(0) = \nabla \underline{f}^T(\underline{x}_k)\underline{d}_k$$

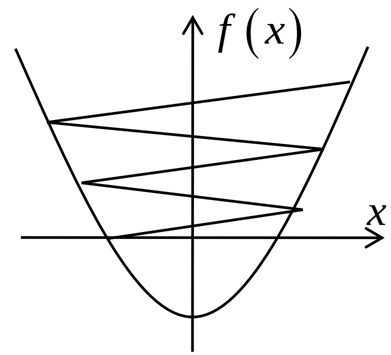
$$g(\alpha) \cong f(\underline{x}_k) + \alpha\nabla \underline{f}^T(\underline{x}_k)\underline{d}_k$$

Counter Example - 1

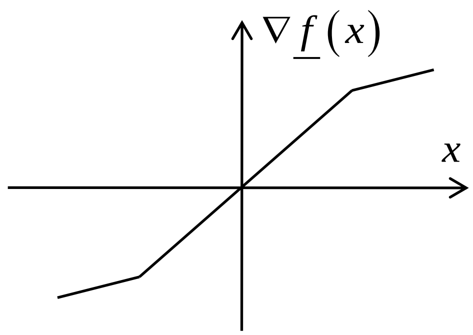
- Example

$$f(x)=\begin{cases}\dfrac{3\left(1-|x|\right)^2}{4}-2\left(1-|x|\right) & \text{if } |x|>1\\ x^2-1 & \text{if } |x|\le 1\end{cases}$$

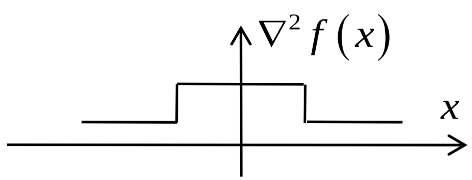
$$\Rightarrow f(x)=\begin{cases}\dfrac{3\left(1-x\right)^2}{4}-2\left(1-x\right) & \text{if } x>1\\ x^2-1 & -1\le x\le 1\\ \dfrac{3\left(1+x\right)^2}{4}-2\left(1+x\right) & \text{if } x<-1\end{cases}$$



$$\nabla f(x)=\begin{cases}2+\dfrac{3}{2}\left(x-1\right) & \text{if } x>1\\ 2x & -1\le x\le 1\\ -2+\dfrac{3}{2}\left(x+1\right) & \text{if } x<-1\end{cases}$$



$$\nabla^2 f(x)=\begin{cases}3/2 & \text{if } x>1\\ 2 & -1\le x\le 1\\ 3/2 & \text{if } x<-1\end{cases}$$



Counter Example - 2

- Suppose $x_0 = 2 \Rightarrow x_0 - \nabla f(x_0) = 2 - 2 - \frac{3}{2} = -\frac{3}{2} \Rightarrow x_1 = -\frac{3}{2} = -\left(1 + \frac{1}{2}\right)$

$$x_2 = x_1 - \nabla f(x_1) = -\frac{3}{2} + 2 + \frac{3}{4} = \frac{5}{4} = \left(1 + \frac{1}{4}\right)$$

$$x_3 = x_2 - \nabla f(x_2) = \frac{5}{4} - 2 - \frac{3}{8} = -\frac{9}{8} = -\left(1 + \frac{1}{8}\right)$$

- Pattern: $2 \quad -\left(1 + \frac{1}{2}\right) \quad \left(1 + \frac{1}{4}\right) \quad -\left(1 + \frac{1}{8}\right) \quad \ldots \quad \left(1 + \frac{1}{2^{2k}}\right) \quad -\left(1 + \frac{1}{2^{2k+1}}\right)$

  converges ± 1 where $\nabla \underline{f}(x) = 2.$ JAMMED!!

- What is happening?
  Although $f(x_{k+1}) < f(x_k)$, the difference $f(x_{k+1}) < f(x_k) \to 0$ as $|x_k| \to 1$

- A possible way out: Need a step size rule that not only guarantees a decrease but also ensures that the function decrease $f(\underline{x}_{k+1}) - f(\underline{x}_k)$ never tends to zero as $\{\underline{x}_k\}$ tends to a nonstationary point (i.e., a nonoptimal point). There are three such rules: Armijo, Goldstein and Wolf's rules. We will discuss the first two here.

Armijo Step Size Rule

- Armijo rule

Given $s > 0,\ \beta \in (0,\ 1),\ \sigma \in \left(0,\ \frac{1}{2}\right)$

Do while

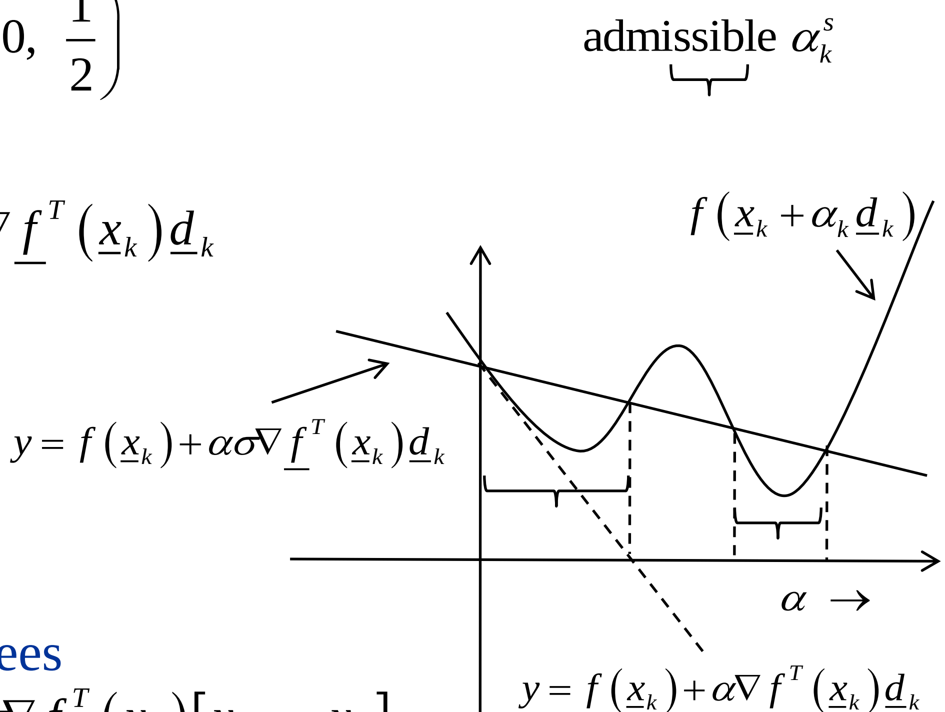
$$f\left(\underline{x}_k + s\underline{d}_k\right) - f\left(\underline{x}_k\right) \geq \sigma s \nabla \underline{f}^T\left(\underline{x}_k\right)\underline{d}_k$$

$$s \leftarrow s\beta$$

Evaluate $f\left(\underline{x}_k + s\underline{d}_k\right)$

end Do

$\alpha_k = s$



- Note that the method guarantees

$$f\left(\underline{x}_k + \alpha_k \underline{d}_k\right) \leq f\left(\underline{x}_k\right) + \sigma\nabla \underline{f}^T\left(\underline{x}_k\right)\left[\underline{x}_{k+1} - \underline{x}_k\right]$$

(or) $\quad f\left(\underline{x}_k + \alpha_k \underline{d}_k\right) \leq f\left(\underline{x}_k\right) + \sigma\alpha_k\nabla \underline{f}^T\left(\underline{x}_k\right)\underline{d}_k$ = UPPER BOUND

Works well in practice, but may result in small α_k

Goldestein Step Size Rule

- Goldstein rule: To guard against small α_k, impose a lower bound on $f(\underline{x}_{k+1})$

Given $s > 0,\ \beta_1 \in (0,\ 1),\ \sigma \in (0,\ 1/2),\ \beta_2 \in (0,\ 1)$

Do until $f(\underline{x}_k) + (1-\sigma) s \nabla \underline{f}^T(\underline{x}_k)\underline{d}_k \le f(\underline{x}_k + s\underline{d}_k) \le f(\underline{x}_k) + \sigma s \nabla \underline{f}^T(\underline{x}_k)\underline{d}_k$

If $f(\underline{x}_k) + \sigma s \nabla \underline{f}^T(\underline{x}_k)\underline{d}_k < f(\underline{x}_k + s\underline{d}_k)$

$s \leftarrow s\beta_1$

else

if $f(\underline{x}_k) + (1-\sigma) s \nabla \underline{f}^T(\underline{x}_k)\underline{d}_k > f(\underline{x}_k + s\underline{d}_k)$

$s \leftarrow s/\beta_2$

end if

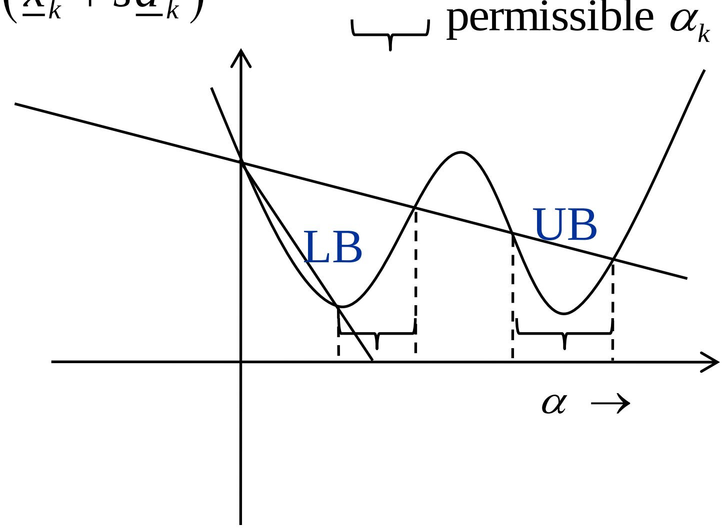
end Do

$\alpha_k = s$

LB: $y = f(\underline{x}_k) + (1-\sigma)\alpha \nabla \underline{f}^T(\underline{x}_k)\underline{d}_k$

UB: $y = f(\underline{x}_k) + \sigma\alpha \nabla \underline{f}^T(\underline{x}_k)\underline{d}_k$

$\boxed{\text{Typical } \sigma = [10^{-5},\ 10^{-1}]}$



Application to Quadratic Function

- Example: quadratic function: $f(\underline{x}) = \frac{1}{2}\underline{x}^T Q \underline{x}\ ; \quad Q > 0$

$$f(\underline{x}_k + \alpha_k \underline{d}_k) = \frac{1}{2}(\underline{x}_k + \alpha_k \underline{d}_k)^T Q(\underline{x}_k + \alpha_k \underline{d}_k); \quad \alpha_k^* = \frac{-\underline{d}_k^T Q \underline{x}_k}{\underline{d}_k^T Q \underline{d}_k}$$

From Goldstein rule: $f(\underline{x}_{k+1}) - f(\underline{x}_k) = \frac{1}{2}(\underline{x}_k + \alpha \underline{d}_k)^T Q(\underline{x}_k + \alpha \underline{d}_k) - \frac{1}{2}\underline{x}_k^T Q \underline{x}_k$

$$= \alpha \underline{d}_k^T Q \underline{x}_k + \frac{\alpha^2}{2}\underline{d}_k^T Q \underline{d}_k$$

$$\Rightarrow \quad (1-\sigma)\alpha \underline{d}_k^T Q \underline{x}_k \le \alpha \underline{d}_k^T Q \underline{x}_k + \frac{\alpha^2}{2}\underline{d}_k^T Q \underline{d}_k \le \sigma \alpha \underline{d}_k^T Q \underline{x}_k$$

Divide by α

$$\Rightarrow \quad (1-\sigma)\underline{d}_k^T Q \underline{x}_k \le \underline{d}_k^T Q \underline{x}_k + \frac{\alpha}{2}\underline{d}_k^T Q \underline{d}_k \le \sigma \underline{d}_k^T Q \underline{x}_k$$

$$-(1-\sigma)\alpha_k^* \le -\alpha_k^* + \frac{\alpha}{2} \le -\sigma\alpha_k^* \quad \Rightarrow \quad 2\sigma\alpha_k^* \le \alpha \le 2(1-\sigma)\alpha_k^*$$

$$\sigma = \frac{1}{2} \quad \Rightarrow \quad \alpha = \alpha_k^*$$

Armijo $\Rightarrow \quad \alpha \le 2(1-\sigma)\alpha_k^*$

Line Search for Optimal Step Size

- Minimize $f(\underline{x}_k + \alpha\, \underline{d}_k)$ with respect to α such that $\alpha \geq 0$

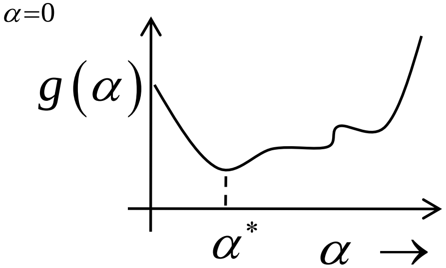
  Pick α_k to minimize $g(\alpha) = f(\underline{x}_k + \alpha\, \underline{d}_k)$, i.e., get the most decrease in the direction $\underline{d}_k$

  From the optimality condition $\left.\dfrac{\partial g(\alpha)}{\partial \alpha}\right|_{\alpha^*} = 0 \Rightarrow \nabla \underline{f}^T\left(\underline{x}_k + \alpha^* \underline{d}_k\right)\underline{d}_k = 0$

- What do we know? $g(0) = f(\underline{x}_k)$; $\left.\dfrac{\partial g(\alpha)}{\partial \alpha}\right|_{\alpha=0} = g'(0) = \nabla \underline{f}^T(\underline{x}_k)\underline{d}_k$

  For Newton type methods, also knows

  $$\left.\frac{\partial^2 g(\alpha)}{\partial \alpha^2}\right|_{\alpha=0} = g''(0) = \underline{d}_k^T \nabla^2 f(\underline{x}_k)\underline{d}_k > 0$$



- We will use $g(0), g'(0)$ and possibly $g''(0)$ to limit the range $(0, \infty)$ for α to a finite range (l_1, r_1) later.
- We assume that $g(\alpha) \rightarrow \infty$ as $\alpha \rightarrow \infty \Rightarrow$ Existence of minimum guaranteed by Weirstrauss' theorem.

Schemes to Find α*

∃ many schemes to find α*. They can be broadly divided into three categories

- Those that use function evaluations only

$$g(\alpha) = f(\underline{x}_k + \alpha\, \underline{d}_k) \rightarrow \begin{cases} \text{Fibonacci search} \\ \text{Golden section search} \\ \text{Quadratic interpolation} \end{cases}$$

Combined Golden section and Quadratic interpolation is the best

- Those that use function and derivative information

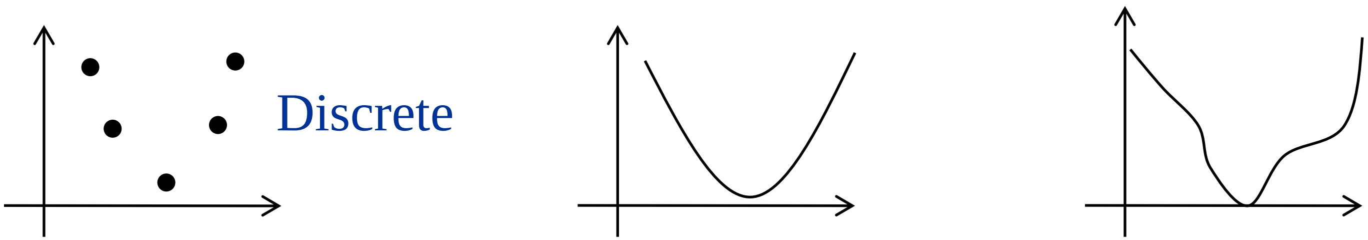
$$g(\alpha) = f(\underline{x}_k + \alpha_k \underline{d}_k);\ g'(\alpha) = \nabla \underline{f}^T(\underline{x}_k + \alpha_k \underline{d}_k)\underline{d}_k \rightarrow \begin{cases} \text{Secant method} \\ \text{Cubic interpolation} \end{cases}$$

- Those that use function, derivative and second derivative info

$\rightarrow$ Newton's method (rarely used for line serach)

Key Idea: Unimodality of Functions

- Assume g (α) is a strictly *unimodal* function on an interval $L_1 = [l_1, r_1]$, i.e., a single global minimum. Unimodal functions need not be smooth.
- Def ^{n}: A function g(α) is unimodal over an interval $L_1 = [l_1, r_1]$ if α_1 and α_2 are two points in L_1 s.t. $\alpha_1 < \alpha_2 < \alpha^*$ or $\alpha^* < \alpha_1 < \alpha_2$. Then
  $g(\alpha_1) > g(\alpha_2) > g(\alpha^*)$ (or) $g(\alpha^*) < g(\alpha_1) < g(\alpha_2)$

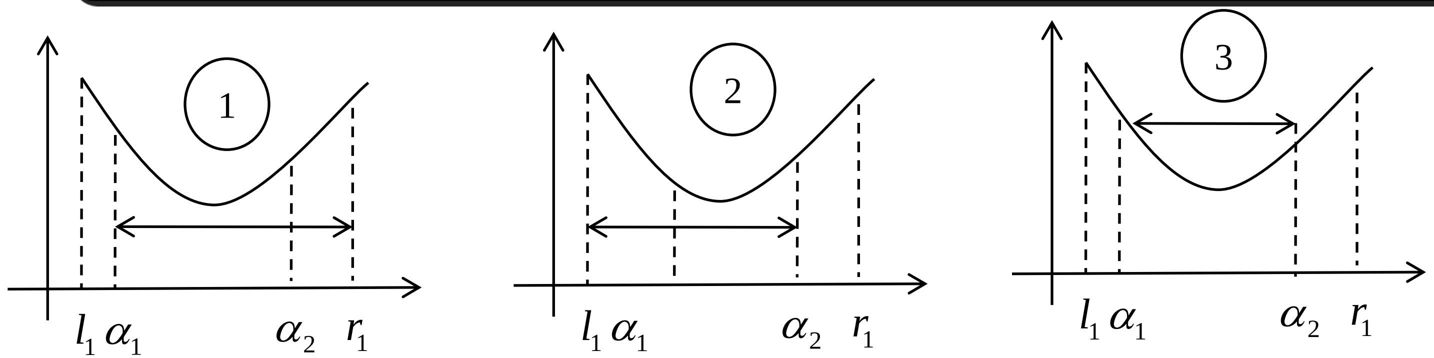


A unimodal function is monotonic on either side of α^*

- Note: a strictly convex function is unimodal
- How to use it?: Suppose $\alpha_1, \alpha_2 \in [l_1, r_1]$ with $\alpha_1 < \alpha_2$ and we find

1. $g(\alpha_1) > g(\alpha_2)$ then $\alpha^* \in [\alpha_1, r_1]$
2. $g(\alpha_1) < g(\alpha_2)$ then $\alpha^* \in [l_1, \alpha_2]$
3. $g(\alpha_1) = g(\alpha_2)$ then $\alpha^* \in [\alpha_1, \alpha_2]$

Bracketing using Unimodal Property



The unimodality assumption helps us to bracket the minimum into smaller and smaller subintervals, e_k

$$[l_1,\ r_1] \rightarrow \left.\begin{matrix} [\alpha_1,\ r_1] \\ [l_1,\ \alpha_2] \\ [\alpha_1,\ \alpha_2] \end{matrix}\right\} (l_2, r_2)$$

$$[l_1,\ r_1] \rightarrow [l_2,\ r_2] \rightarrow \ldots \rightarrow [l_N,\ r_N]$$

$$e_1 \rightarrow e_2 \rightarrow \ldots \rightarrow e_N$$

$$e_1 > e_2 > \ldots > e_k \approx \text{TOL}$$

- Suppose want to find the minimum sized interval $[l_1, r_1]$ using N function evaluations. Also want to evaluate function only once in going from one interval to the next

$$[l_k,\ r_k] \underset{\textit{one function evaluation}}{\longrightarrow} [l_{k+1},\ r_{k+1}]$$

Fibonacci Search - 1

- We can accomplish this using Fibonacci search first discovered by Leonardo of Pisa (1202). Independent of $g(\alpha)$ as long as $g(\alpha)$ is unimodal
  - If $N = 2$, pick α_1 and α_2 at center and close to each other

$$l_1 \qquad \alpha_1 = \frac{l_1 + r_1}{2} - \varepsilon \qquad \alpha_2 = \frac{l_1 + r_1}{2} + \varepsilon \qquad r_1$$

$$[l_1,\ r_1] \to \begin{cases} \left[l_1,\ \frac{l_1 + r_1}{2} + \varepsilon\right] & \text{if } g(\alpha_1) < g(\alpha_2) \\ \left[\frac{l_1 + r_1}{2} - \varepsilon,\ r_1\right] & \text{if } g(\alpha_1) > g(\alpha_2) \end{cases}$$

$$e_2 \approx \frac{e_1}{2}$$

  - If $N = 3$ … two step process

$\boxed{\text{Note: } \alpha_1 \text{ or } \alpha_2 \text{ are in place}}$

$$l_1 \qquad \alpha_1 = l_1 + \frac{r_1 - l_1}{3} \qquad \alpha_2 = l_1 + \frac{2(r_1 - l_1)}{3} \qquad r_1$$

First isolate which $\frac{2}{3}$ of interval α^* lies in

$[l_1,\ \alpha_2]$ or $[\alpha_1,\ r_1]$

Then, reduce $[l_1,\ \alpha_2] \sim [\alpha_1,\ r_1]$ by $\frac{1}{2}$

$$\Rightarrow \quad e_3 = \frac{e_2}{2} = \frac{2}{3} \cdot \frac{1}{2} \cdot e_1 = \frac{e_1}{3}$$

Fibionacci Search - 2

- If $N = 4$…. 3 step process

$l_1 \qquad l_1 + \frac{2(r_1 - l_1)}{5} \qquad l_1 + \frac{3(r_1 - l_1)}{5} \qquad r_1$

$l_1 \quad l_1 + \frac{r_1 - l_1}{5} \quad l_1 + \frac{2(r_1 - l_1)}{5} \quad l_1 + \frac{3(r_1 - l_1)}{5}$

$$\boxed{e_2 = \frac{3}{5} e_1}$$

$l_1 \qquad l_1 + \frac{2(r_1 - l_1)}{5}$

$$\boxed{\begin{aligned} e_3 &= \frac{2}{3} e_2 = \frac{2}{5} e_1 \\ e_4 &= \frac{1}{2} e_3 = \frac{1}{5} e_1 \end{aligned}}$$

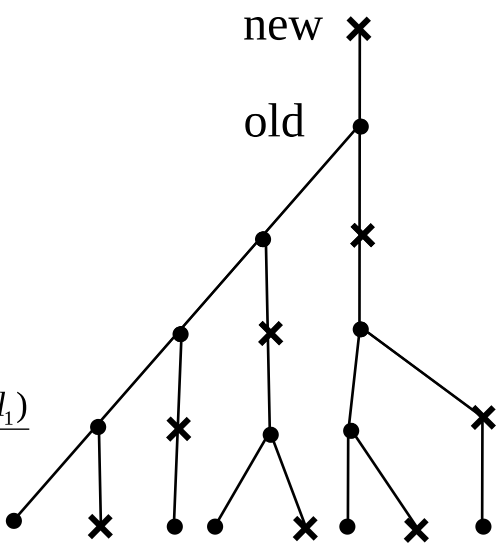
$l_1 \qquad l_1 + \frac{(r_1 - l_1)}{5}$ Evaluate to left or right of $l_1 + \frac{(r_1 - l_1)}{5}$

N	2	3	4
e_N	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{5}$

$$\Rightarrow F_k = F_{k-1} + F_{k-2}$$

Rabbits

- One month to fertility
- One month to produce pair
- Never die



$F_0 = 1$
$F_1 = 1$
$F_2 = 2$
$F_3 = 3$
$F_4 = 5$
$F_5 = 8$

Fibionacci Search - 3

- For general N, let k = 1, 2, .., N–1. The Fibionacci method uses at step k

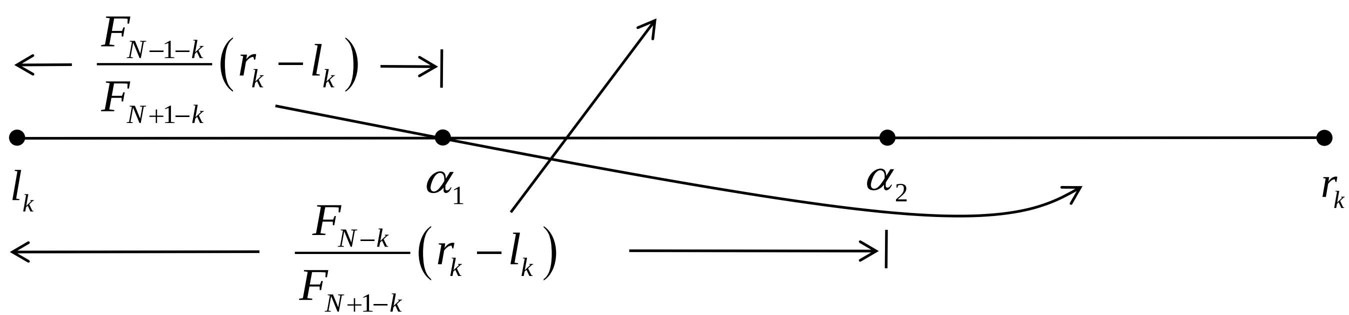
$$\alpha_1 = l_k + \frac{F_{N-1-k}}{F_{N+1-k}}(r_k - l_k)$$

$$\alpha_2 = l_k + \frac{F_{N-k}}{F_{N+1-k}}(r_k - l_k)$$

where except for k = 1, one of the points α_1 or α_2 was already evaluated at a previous iteration.

where F_k are the Fibionacci numbers

$$F_k = F_{k-1} + F_{k-2}; \quad F_0 = F_1 = 1$$



SYMMETRIC INTERVALS

Interval Reduction for Fibionacci Search

$$r_k - \alpha_2 = r_k - l_k - \frac{F_{N-k}}{F_{N+1-k}}(r_k - l_k) = (r_k - l_k)\frac{F_{N-k-1}}{F_{N+1-k}} = \alpha_1 - l_1$$

where for $k = N-1$, the last point is at a place close the remaining one, because

$$\alpha_1 = l_{N-1} + \frac{1}{2}(r_{N-1} - l_{N-1}) \quad ; \quad \alpha_2 = l_{N-1} + \frac{1}{2}(r_{N-1} - l_{N-1})$$

Basic result: $r_N - l_N = \dfrac{r_1 - l_1}{F_N} + \varepsilon$ smallest possible reduction

and $r_k - l_k = \dfrac{F_{N-k+1}}{F_N}(r_1 - l_1)$

- Proof:

$$r_{k+1} - l_{k+1} = \frac{F_{N-k}}{F_{N+1-k}}(r_k - l_k)$$

$$\Rightarrow \; r_2 - l_2 = \frac{F_{N-1}}{F_N}(r_1 - l_1); r_3 - l_3 = \frac{F_{N-2}}{F_{N-1}} \cdot \frac{F_{N-1}}{F_N}(r_1 - l_1) = \frac{F_{N-2}}{F_N}(r_1 - l_1)$$

$$\Rightarrow \; r_k - l_k = \frac{F_{N-k+1}}{F_N}(r_1 - l_1) \Rightarrow r_N - l_N = \frac{F_1}{F_N}(r_1 - l_1) = \frac{1}{F_N}(r_1 - l_1)$$

Advantages and Disadvantages

- Advantages of Fibionacci method:
  - Smallest interval for a given N
  - No derivative information
  - Simple to implement
- Disadvantages of Fibionacci method:
  - Need to prespecify N. Also need to compute (or store) Fibionacci numbers. Luckily, don't have to if N is large. Leads to GOLDEN SECTION SEARCH
- Golden section search retains

  a) Symmetry of interval reduction

  b) Only one function evaluation per step

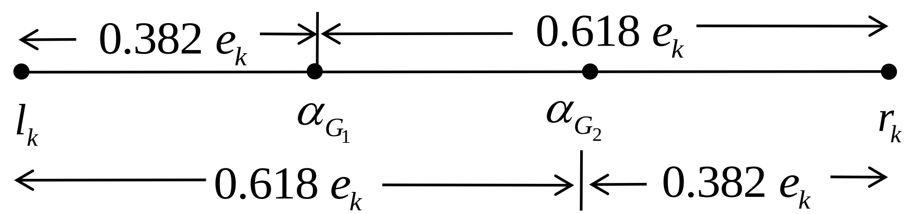
Consider the limiting ratio of $\frac{F_N}{F_{N-1}} \triangleq \tau$ as $N \to \infty$

$$F_N = F_{N-1} + F_{N-2} \quad \Rightarrow \quad \frac{F_N}{F_{N-1}} = 1 + \frac{F_{N-2}}{F_{N-1}}$$

As $N \to \infty \quad \Rightarrow \quad \tau = 1 + \frac{1}{\tau} \quad \Rightarrow \quad \tau^2 - \tau - 1 = 0 \;\; \text{(or)} \;\; \tau = 1.618$

Golden Section Search

- Note: $\frac{1}{\tau} = \tau - 1 = 0.618$



$e_k = r_k - l_k$

$$e_{k+1} \begin{cases} \alpha_{G_2} - l_k = 0.618 e_k = \frac{1}{\tau} e_k \\ r_k - \alpha_{G_1} = 0.618 e_k = \frac{1}{\tau} e_k \end{cases}$$

$$\Rightarrow \quad (e_N)_k = \frac{1}{\tau^{N-1}} e_1$$

$$e_{k+1} = \frac{1}{\tau} e_k$$

order of convergence: 1

convergence ratio: $\beta = \frac{1}{\tau} = 0.618$

Note that α_{G_1} or α_{G_2} is in the correct position for the next step.

since $\boxed{\frac{1}{\tau^2} = 1 - \frac{1}{\tau}}$

Golden Section vs. Fibionacci Search

- Golden section method yields 17% larger interval than Fibionacci for the same $N \Rightarrow \lim_{N\to\infty} \frac{F_N}{\tau^{N-1}} = 1.17$

Characteristic Equation : $z^2 - z - 1 = 0$

$$\Rightarrow z = \frac{1 \pm \sqrt{5}}{2}$$

$$z_1 = \frac{1+\sqrt{5}}{2} = \tau, \quad z_2 = \frac{1-\sqrt{5}}{2} = \frac{-1}{\tau}$$

- Proof:

$$F_k = F_{k-1} + F_{k-2}$$

$$F_k = A\tau_1^k + B\left(\frac{-1}{\tau}\right)^k$$

$$F_0 = F_1 = 1 \Rightarrow \left.\begin{aligned} A + B &= 1 \\ A\tau - \frac{B}{\tau} &= 1 \end{aligned}\right\} \Rightarrow \begin{aligned} A &= \frac{1 + 1/\sqrt{5}}{2} \\ B &= \frac{1 - 1/\sqrt{5}}{2} \end{aligned}$$

$$F_N = \left[\tau^{N+1} - \left(\frac{-1}{\tau}\right)^{N+1}\right] \Big/ \sqrt{5}$$

$$\lim_{N\to\infty} \frac{F_N}{\tau^{N-1}} = \frac{\tau^2}{\sqrt{5}} = 1.17$$

Golden section search is 17% less efficient than Fibionacci search, but is much easier to implement.

Summary

- Reviewed necessary and sufficient conditions of optimality
- Gradients and Contour Maps
  - Gradient is orthogonal to contour curves
  - Descent directions
- Algorithms for Unconstrained Minimization
  - Steepest descent, Diagonally scaled steepest descent, Newton, Discretized Newton, Gauss-Newton, Conjugate gradient, Quasi-Newton
- Step Size Rules (or) Line Search Methods
  - Armijo, Goldstein, Fibionacci search and Golden section search
  - Next: combine Golden section with quadratic interpolation