# Intelligent Signal Processing and Control

# Mathematical Foundations IV

Aurobinda Routray

# Optimization

# Optimization: an Overview

$$\underset{\mathbf{x}}{\text{minimize}} \quad E(\mathbf{x})$$

# Unconstrained Optimization

# Steepest Descent Method

$E(\mathbf{x})$ to be minimized

$\mathbf{g}_k$ is the gradient at $k^{th}$ step

$\mathbf{g}_k = \nabla E(\mathbf{x}_k)$

$\mathbf{x}_{\mathrm{k+1}} = \mathbf{x}_k - \alpha \mathbf{g}_k$

$\alpha$ is an **unknown** non-negative constant that minimizes $E(\mathrm{x})$

# Stability

$$\frac{d\mathbf{x}(t)}{dt} = -\boldsymbol{\mu}(\mathbf{x},t)\nabla_x E(\mathbf{x})$$

$$\frac{dE(t)}{dt} = \frac{dE(t)}{d\mathbf{x}}\frac{d\mathbf{x}}{dt} = -\nabla_x^T E(\mathbf{x})\boldsymbol{\mu}(\mathbf{x},t)\nabla_x E(\mathbf{x}) < 0$$

only when $\boldsymbol{\mu}(\mathbf{x},t)$ is positive definite

$\boldsymbol{\mu}(\mathbf{x},t)$ is known as the learning matrix

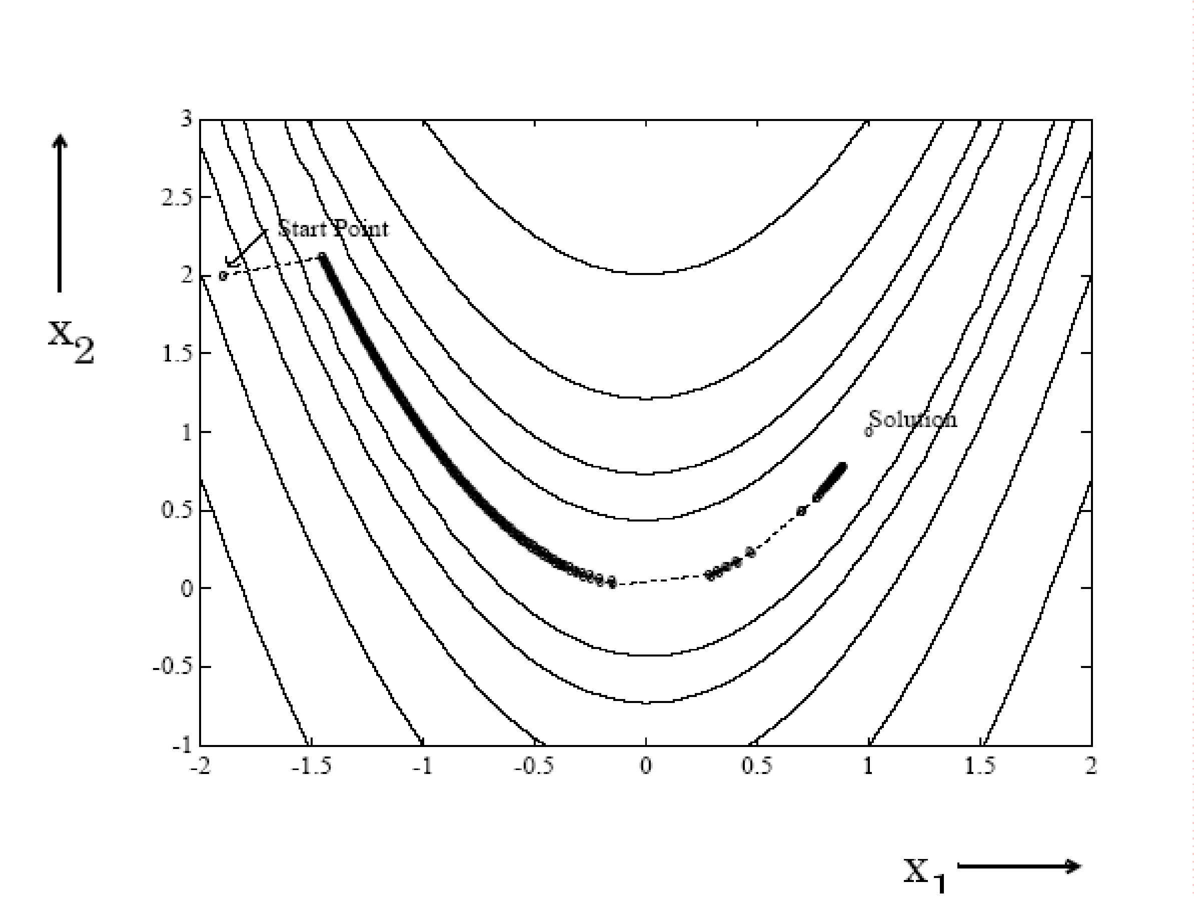
can be assumed to be $\mu.\mathbf{I}$

where $\mu$ is a scalar and $\mathbf{I}$ is an identity matrix

$\mu$ can be adaptively adjusted within $[\mu_{\min}, \mu_{\max}]$

# Pit falls

**Rosenbrock's Function**



# Newton's Methods

**Newton's method does local approximation of the**

**Function $\varepsilon$ to be minimized in the form of a quadratic function**

$$q(x) = \varepsilon(x_k) + \varepsilon'(x_k)(x - x_k) + \frac{1}{2}\varepsilon''(x_k)(x - x_k)^2$$

$$\dot{q}(x) = \frac{dq(x)}{dx} = \varepsilon'(x_k) + \varepsilon''(x_k)(x - x_k)\Big|_{\mathbf{x}=\mathbf{x}_{k+1}} = 0$$

Which gives

$$x_{k+1} = x_k - \frac{\varepsilon'(x_k)}{\varepsilon''(x_k)}$$

For multivariable functions of the form $\varepsilon(x_1, x_2, x_3, \ldots .x_n)$

Near $\mathbf{x_k}$ the Taylor's series expansion is given as

$$\varepsilon(\mathbf{x}) \cong \varepsilon(\mathbf{x_k}) + \nabla_x^T \varepsilon(\mathbf{x_k})(\mathbf{x} - \mathbf{x_k}) + \frac{1}{2}(\mathbf{x} - \mathbf{x_k})^T \mathrm{H}_k (\mathbf{x} - \mathbf{x_k})$$

$$\nabla_{\mathbf{x}} \varepsilon(\mathbf{x})\Big|_{\mathbf{x}=\mathbf{x}_{k+1}} = 0$$

*at minimum* **H** *should be positive definite*

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathrm{H}^{-1} \nabla_x \varepsilon(\mathbf{x}_k)$$

**OR**

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}^{-1} \mathbf{g}_k}$$

# Quasi Newton's Methods

To maintain the positive definiteness of the matrix modifications of the Hessian is necessary

$$\mathbf{x}_{k+1} = \mathbf{x}_{k} - \alpha_{k} \mathbf{H}^{-1} \mathbf{g}_{k}$$

$\alpha_k$ **is a search parameter to minimize** $\varepsilon$

Perturb **H** to prevent ill conditioning by ***LDU*** decomposition

$$\mathbf{H} = \mathbf{LDU} = \mathbf{LDL}^{T}$$

The smallest diagonal element of **D** should be increased to prevent singularity

# Algorithm **Modified Newton's Method**

Step 1. Select an initial solution vector $\mathbf{x}_0$ and convergence tolerance *epsi*

Step 2. For k=0,1,2& compute $\mathbf{g_k} = \nabla \boldsymbol{\varepsilon}\left[\mathbf{x_k}\right]$ if $\left\|\mathbf{g_k}\right\| < epsi$ stop

Step 3. Compute $\mathbf{H} = \mathbf{L.D.L^T}$

Step 4. Modify the diagonal elements $\mathbf{D} \leftarrow \mathbf{D_m}$

Step 5. Compute the search direction from $\left(\mathbf{LD_m L^T}\right)\mathbf{d_k} = -\mathbf{g_k}$

Step 6. Perform line search to determine

$\mathbf{x_{k+1}} = \mathbf{x_k} + \alpha_k \mathbf{d_k}$ the new solution estimate where $\alpha_k$ is selected

such that $\min_{\alpha \geq 0} \mathcal{E}\left(\mathbf{x_k} + \alpha \mathbf{d_k}\right)$

# Algorithm Broyden-Goldfarb-Shanno (BFGS) algorithm

Step 1. Select an initial solution vector $\mathbf{x}_0$ and initial Hessian approximation $\mathbf{B_0} = \mathbf{I}$

Step 2. For k=0,1,2 if $\mathbf{x}_k$ is optimal in some sense then stop

Step 3. Else compute the gradient of the objective function that is $\mathbf{g}_k = \nabla \boldsymbol{\varepsilon}\left[\mathbf{x}_k\right]$

then solve $\mathbf{B}_k \mathbf{d}_k = -\mathbf{g}_k$ for $\mathbf{d}_k$

Step 4. Perform line search to determine

$\mathbf{x_{k+1}} = \mathbf{x_k} + \alpha_k \mathbf{d_k}$ the new solution estimate where $\alpha_k$ is selected

such that $\min\limits_{\alpha \ge 0} \mathcal{E}\left(\mathbf{x_k} + \alpha \mathbf{d_k}\right)$

Step 5. Compute $\boldsymbol{\delta}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ and $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$

Step 6. Compute

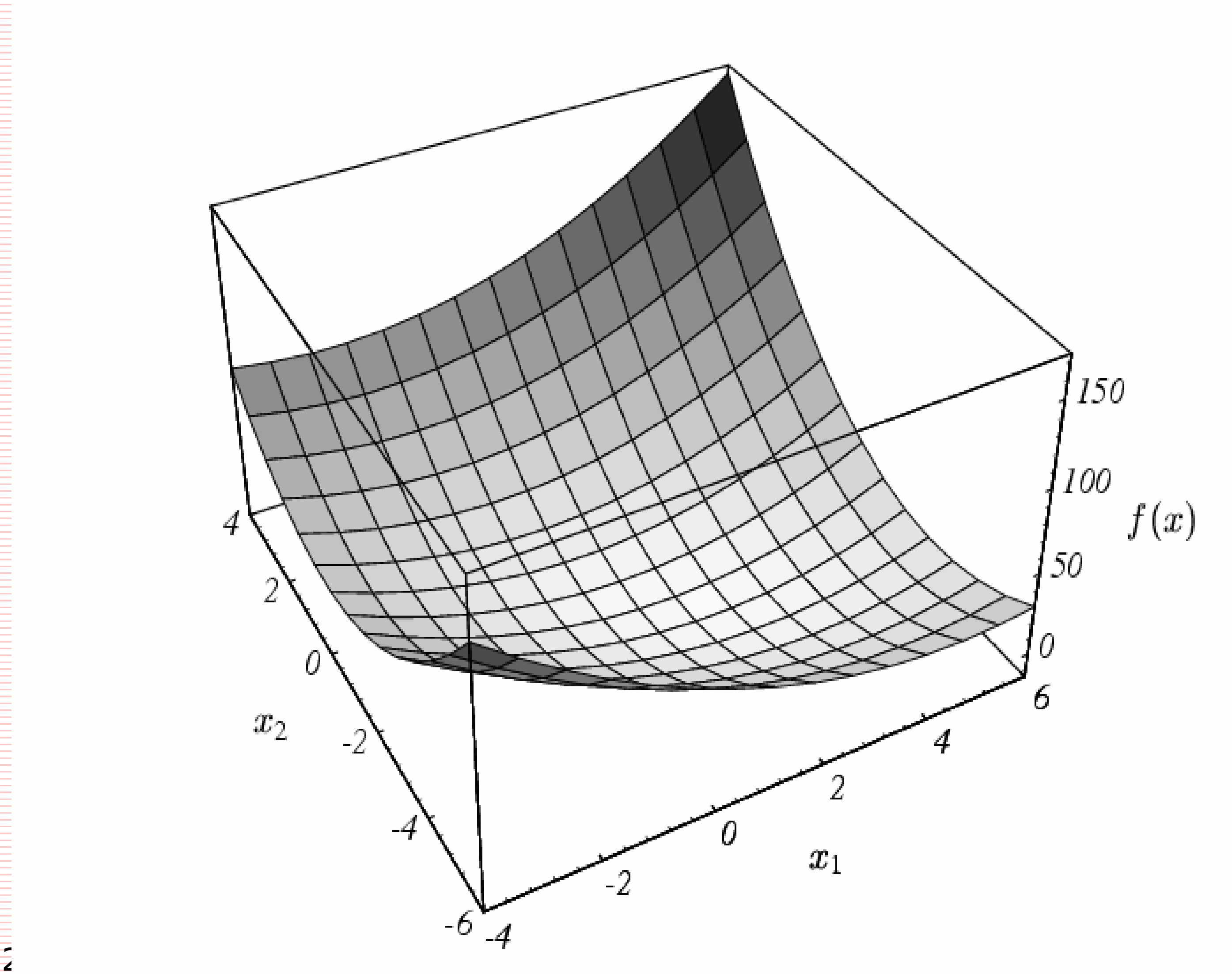
$$\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{\left(\mathbf{B}_k \boldsymbol{\delta}_k\right)\left(\mathbf{B}_k \boldsymbol{\delta}_k\right)^T}{\boldsymbol{\delta}_k^T \mathbf{B}_k \boldsymbol{\delta}_k} + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \boldsymbol{\delta}_k}$$

where $\mathbf{B}_k$ is the current estimate of the Hessian $\nabla_{\mathbf{x}}^{\mathbf{2}} \mathcal{E}\left(\mathbf{x}_k\right)$

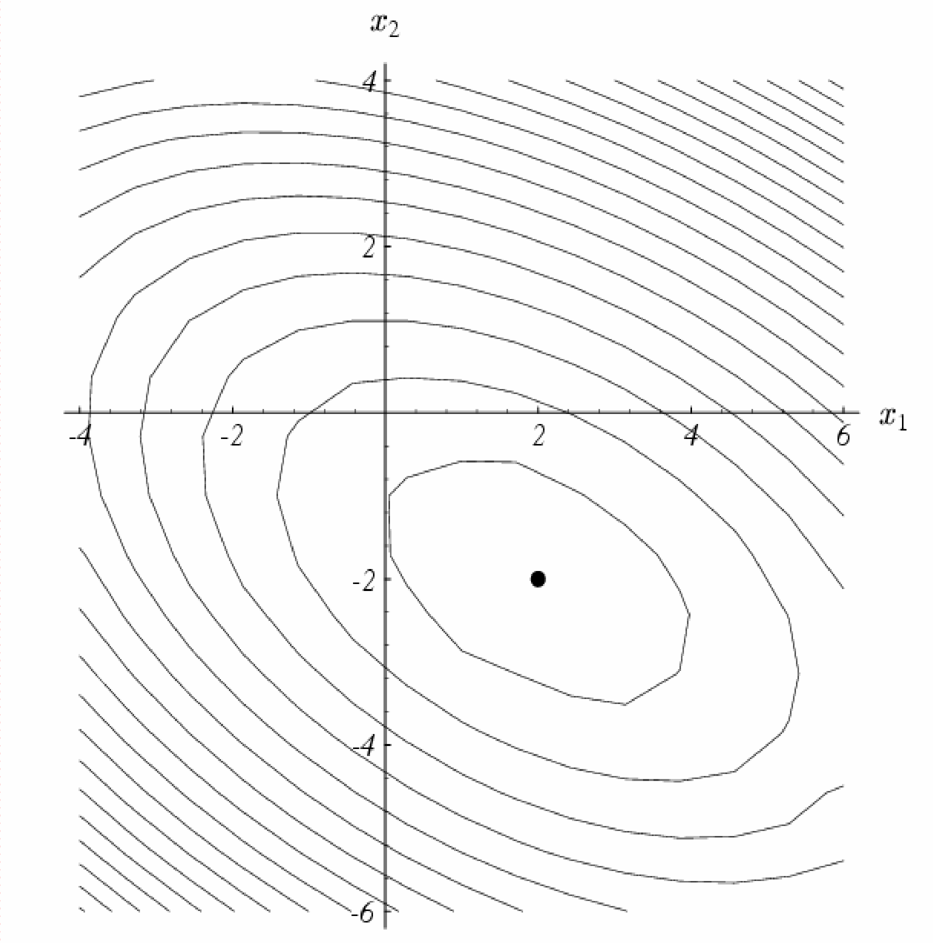
Step 7. Go to Step 2.

# Conjugate Gradient Algorithm

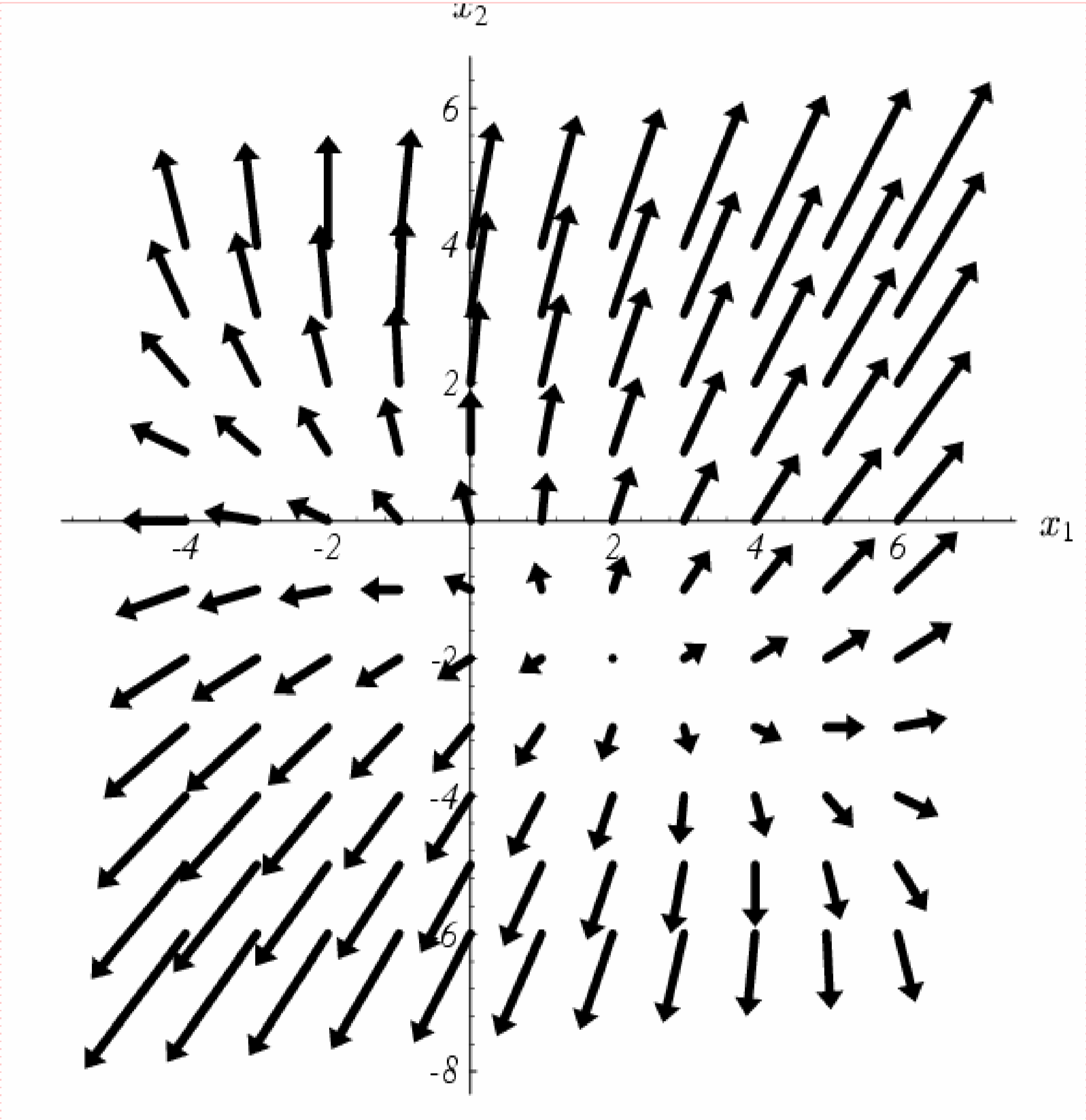
$\mathbf{Qx} = \mathbf{b}$ **solving it is minimizing** $\mathcal{E}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{Qx} - \mathbf{x}^{\mathrm{T}}\mathbf{b}$



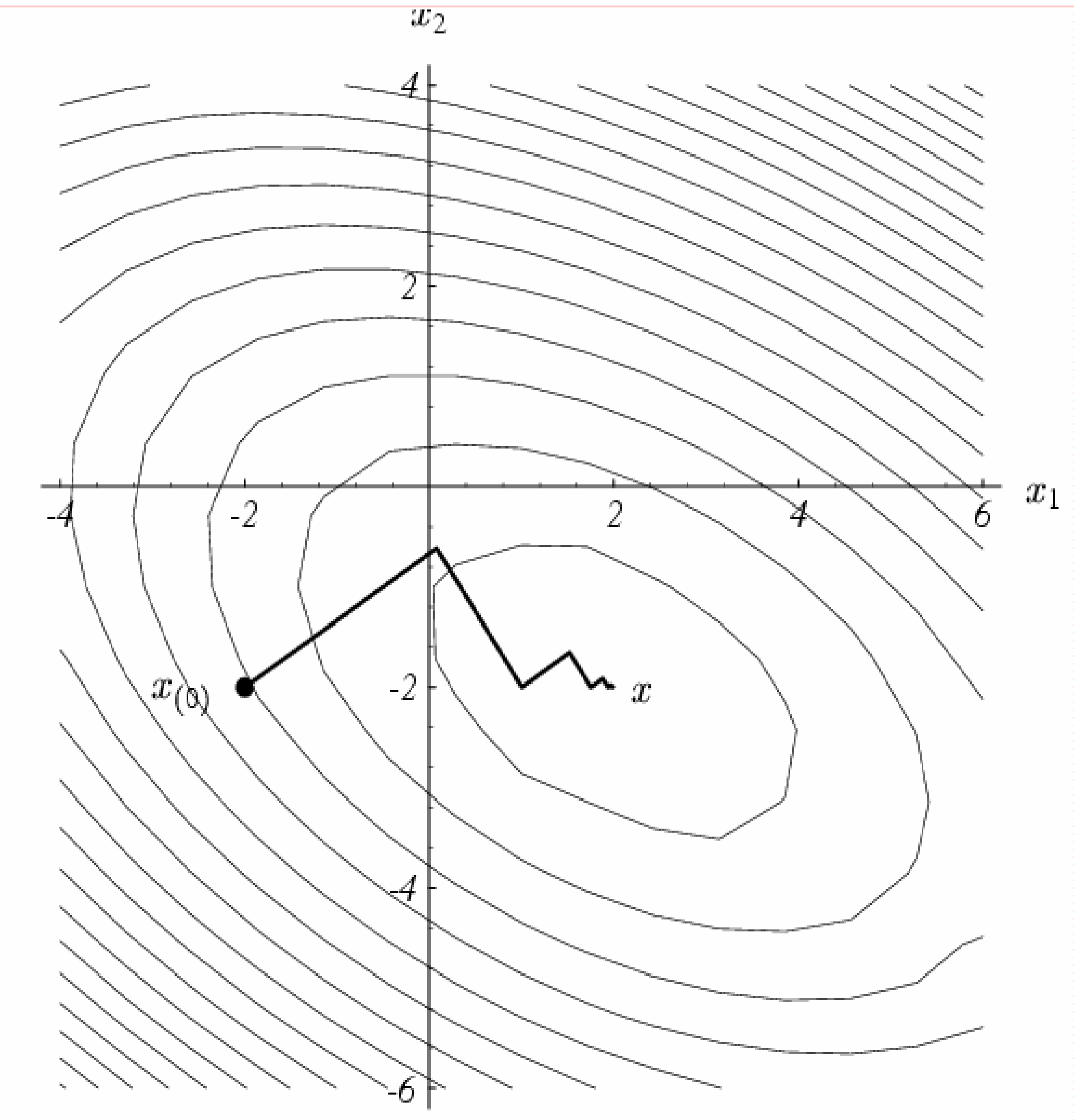
# The Contours



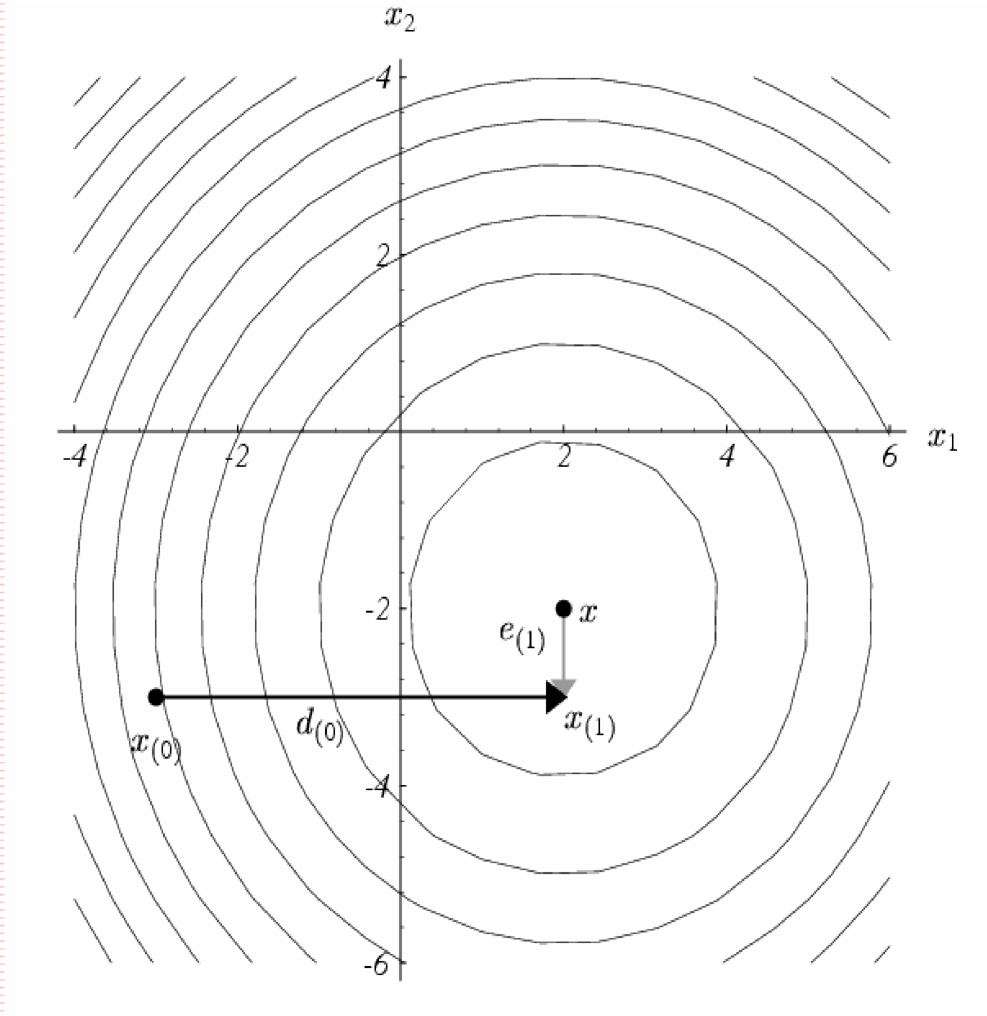
# The Gradient



# Directions of Steepest Descent



# Conjugate Directions



To find conjugate directions

$$J(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathbf{T}}\mathbf{A}\mathbf{x} - \mathbf{b}^{\mathbf{T}}\mathbf{x}$$

$$\mathbf{x}_{\mathbf{F}} = \min_{\mathbf{x}} J(\mathbf{x})$$

$$\mathbf{e}_i = \mathbf{x}_{\mathbf{i}} - \mathbf{x}_{\mathbf{F}}$$

$$\mathbf{x}_{i+1} = \mathbf{x}_{\mathbf{i}} + \alpha_i \mathbf{d}_i$$

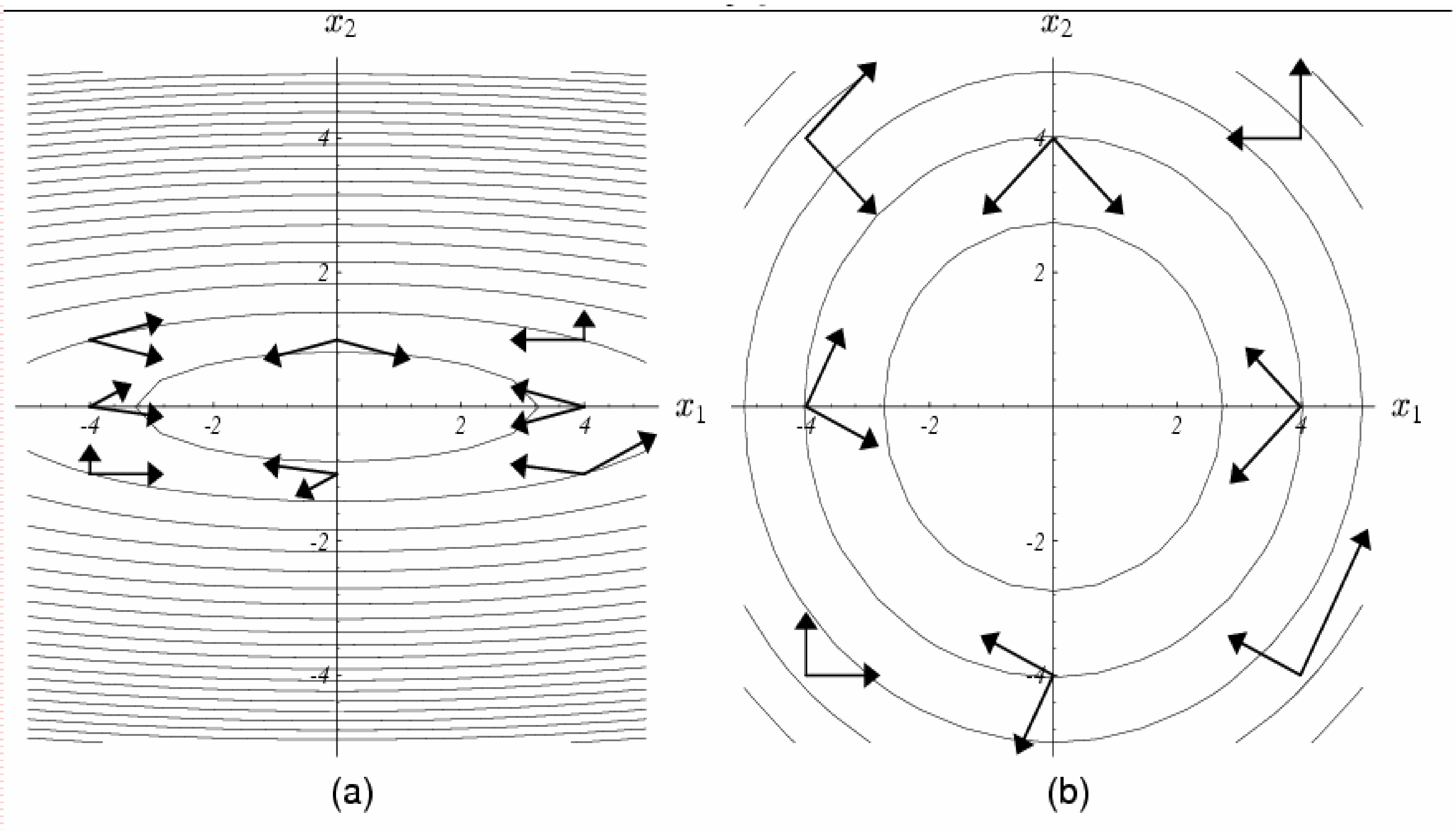
$$\mathbf{d}_i^T \mathbf{e}_{i+1} = 0 \quad \text{for conjugate direction}$$

$$\mathbf{e}_{i+1} = \mathbf{x}_{\mathbf{i+1}} - \mathbf{x}_{\mathbf{F}} = \mathbf{x}_{\mathbf{i}} + \alpha_i \mathbf{d}_i - \mathbf{x}_{\mathbf{F}} = \mathbf{e}_{\mathbf{i}} + \alpha_i \mathbf{d}_i$$

$$\alpha_i = -\frac{\mathbf{d}_i^T \mathbf{e}_i}{\mathbf{d}_i^T \mathbf{d}_i} \quad \text{but } \mathbf{e}_{\mathbf{i}} \text{ is not known}$$

Therefore instead of making the directions

$\mathbf{e}$ othorgonal make it $\mathbf{A}$ orthogonal



$x_2$
$x_1$
4
2
-2
-4
(a)
(b)

$$\mathbf{x}_{i+1} = \mathbf{x_i} + \alpha_i \mathbf{d}_i$$

$$\frac{dJ(\mathbf{x}_{i+1})}{d\alpha} = 0$$

$$J^{'}\left(\mathbf{x}_{i+1}\right)\frac{d\mathbf{x}_{i+1}}{d\alpha} = 0$$

$$-\mathbf{r}_{i+1}^{T}\mathbf{d_i} = 0 \qquad \left[\because \mathrm{r}_{\mathrm{i+1}} = \mathbf{b} - \mathbf{A}\mathbf{x}_{\mathbf{i+1}}\right]$$

$$-\mathbf{r}_{i+1}^{T}\mathbf{d_i} = 0$$

$$\mathbf{d_i^T A e_{i+1}} = 0$$

$$\left[\because \mathbf{b} - \mathbf{A x_F} = 0 \rightarrow \mathbf{b} = \mathbf{A x_F} \rightarrow \mathbf{r_{i+1}} = \mathbf{b} - \mathbf{A x}_{i+1} \rightarrow \mathbf{A}\left(\mathbf{x_F} - \mathbf{x_{i+1}}\right)\right]$$

$$\mathbf{d_i^T A}\left(\mathbf{x_{i+1}} - \mathbf{x_F}\right) = 0$$

$$\mathbf{d_i^T A}\left(\mathbf{x_i} + \alpha \mathbf{d_i} - \mathbf{x_F}\right) = 0$$

$$\alpha = \frac{\mathbf{d_i^T r_i}}{\mathbf{d_i^T A d_i}}$$

## Gram-Schmidt Conjugation

All that is needed now is a set of A-orthogonal search directions

$$\mathbf{u_0}, \mathbf{u}_1, \cdots\cdots, \mathbf{u}_n \text{ n independent vectors}$$

$$d_{(i)} = u_i + \sum_{k=0}^{i-1} \beta_{ik} d_{(k)},$$

where

$\beta_{\text{ik}}$ is defined for $i > k$

# Algorithm **Fletcher-Reeves Conjugate Gradient Method**

Step 1. start with any $\mathbf{x}_0 \in \mathbb{R}^{n\times 1}$

Step 2. Compute $\mathbf{g}_0 = \nabla_x \mathcal{E}(\mathbf{x}_k)\big|_{k=0}$

Step 3. $\mathbf{d}_0 = -\mathbf{g}_0$

Step 4. $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$ such that $\min_{\alpha\geq 0} \mathcal{E}\left(\mathbf{x_k} + \alpha \mathbf{d_k}\right)$

Step 5. $\mathbf{g}_k = \nabla_x \mathcal{E}(\mathbf{x}_{k+1})$

Step 6. $\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k$ and $\beta_k = \dfrac{\mathbf{g}_{k+1}^T \mathbf{d}_k}{\mathbf{d}_k^T \mathbf{d}_k}$

steps 4 through 6 are carried out for $k = 0, 1, \ldots\ldots, n\text{-}1$

Step 7. Replace $\mathbf{x}_0$ by $\mathbf{x}_n$ and go to step 1

Step 8. Continue untill convergence is achieved. Termination criterion is

$$\|\mathbf{d}_k\| < \varepsilon$$

# Conjugate Gradient Algorithm

$$\mathbf{Qx} = \mathbf{b} \text{ solving it is minimizing } \mathcal{E}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{Qx} - \mathbf{x}^{\mathrm{T}}\mathbf{b}$$

## Algorithm **Conjugate Gradient Method**

Step 1. start with any $\mathbf{x}_0 \in \mathbb{R}^{n\times 1}$. define the initial vector as

$$\mathbf{d}_0 = -\mathbf{g}_0 = -\nabla_x \mathcal{E}(\mathbf{x}_k)\big|_{k=0} = \mathbf{b} - \mathbf{Qx}_0$$

Step 2. $\alpha_k = -\dfrac{\mathbf{g}_k^{\ T}\mathbf{d}_k}{\mathbf{d}_k^{\ T}\mathbf{Qd}_k}$ where $\mathbf{g}_k = \mathbf{Qx}_k - \mathbf{b}$

Step 3. $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$

Step 4. $\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k$ and $\beta_k = \dfrac{\mathbf{g}_{k+1}^T \mathbf{Qd}_k}{\mathbf{d}_k^T \mathbf{Qd}_k}$

an alternate form is $\beta_k = \dfrac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_k}$

Step 5. Go to step 2

# Constrained Optimization

$$\text{minimize } f_0(\mathbf{x})$$

$$\text{subject to } f_i(\mathbf{x}) \le 0 \qquad i = 1, 2, \cdots, m$$

$$h_i(\mathbf{x}) = 0 \qquad i = 1, 2, \cdots, p$$

$$\mathbf{x} \in \mathbf{R}^n \qquad D = \bigcap_{i=1}^{m} \mathbf{dom}\, f_i \cap \bigcap_{i=1}^{p} \mathbf{dom}\, h_i \text{ is nonempty}$$

let the optimal value is $p^*$

Define Lagrangian associated with the above problem as

$$L : \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$$

$$L(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{v}) = f_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) + \sum_{i=1}^{h} \nu_i h_i(\mathbf{x})$$

$\lambda_i$ and $\nu_i$ are dual variables and the **dual function** is

$$g(\boldsymbol{\lambda}, \mathbf{v}) = \inf_{\mathbf{x} \in \mathbf{D}} L(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{v}) = \inf_{\mathbf{x} \in \mathbf{D}} \left[ f_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) + \sum_{i=1}^{h} \nu_i h_i(\mathbf{x}) \right]$$

For dual variables $\lambda_i$ and $\nu_i$ **positive**

the dual problems always gives values less than $p^*$

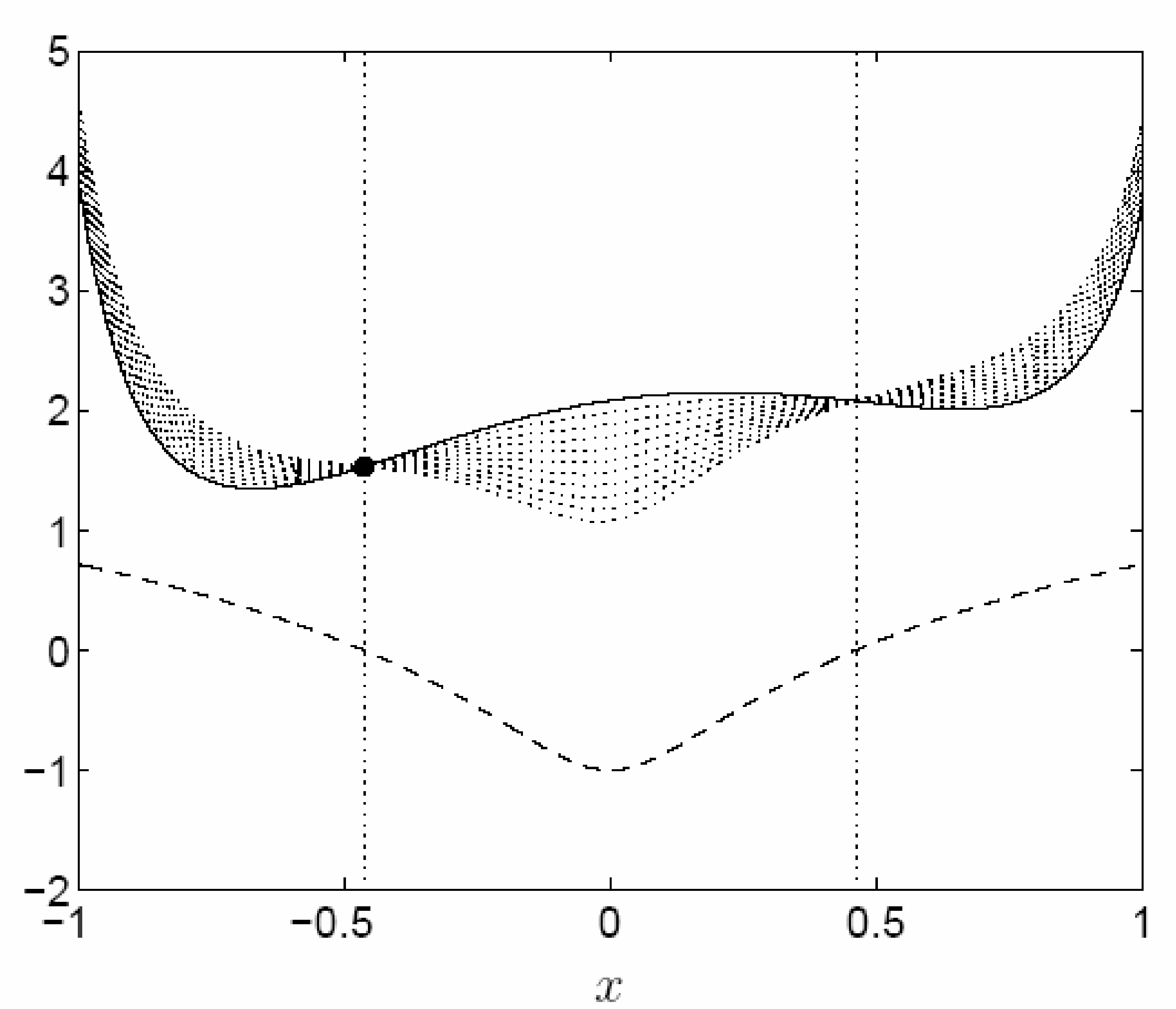
$$g(\lambda, \nu) \le p^*$$

Suppose $\tilde{\mathbf{x}}$ is a feasible point then

$$\sum_{i=1}^{m} \lambda_i f_i(\tilde{\mathbf{x}}) + \sum_{i=1}^{h} \nu_i h_i(\tilde{\mathbf{x}}) \le 0$$

$$L(\tilde{\mathbf{x}}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f_0(\tilde{\mathbf{x}}) + \sum_{i=1}^{m} \lambda_i f_i(\tilde{\mathbf{x}}) + \sum_{i=1}^{h} \nu_i h_i(\tilde{\mathbf{x}}) \le f_0(\tilde{\mathbf{x}})$$

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\mathbf{x} \in \mathbf{D}} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) \le L(\tilde{\mathbf{x}}, \boldsymbol{\lambda}, \boldsymbol{\nu}) \le f_0(\tilde{\mathbf{x}})$$




Lower bound from a dual feasible point. The solid curve shows the objective function $f_0$, and the dashed curve shows the constraint function $f_1$. The feasible set is the interval [-0.46, 0.46] indicated by two dotted vertical lines. The optimal value is at -0.46, $p^*$=1.54

# Lagrange's Dual Problem

To find out $p^*$ is to find the upper bound of $g(\lambda,\nu)$

$$\text{maximize } g(\lambda,\nu)$$
$$\lambda > 0$$

say the maximum value $g(\lambda,\nu)$ is $d^*$

Then always $d^* \leq p^*$ $\left[\text{for convex problems equality holds } \textbf{"Strong Duality"}\right]$

$\left(p^* - d^*\right)$ is knowns as the duality gap

# Example: Least Square Solution

$$\text{minimize} \qquad \mathbf{x}^T\mathbf{x}$$

$$\text{subject to} \qquad \mathbf{A}\mathbf{x} = b$$

Augmented function

$$\text{minimize } L(\mathbf{x}, \nu) = \mathbf{x}^T\mathbf{x} + \nu(A\mathbf{x} - b)$$

$$\nabla L(\mathbf{x}, \nu) = 2\mathbf{x} + \mathbf{A}^T\nu = 0$$

$$\mathbf{x} = -\frac{1}{2}\mathbf{A}^{\mathbf{T}}\nu$$

Dual Function

$$\mathrm{g}(\nu) = L\left(-\left(\frac{1}{2}\right)A^T\nu, \nu\right) = -\left(\tfrac{1}{4}\right)\nu^T AA^T\nu - b^T\nu$$

# KKT [Karush-Kuhn-Tucker Conditions]

Assume the functions $f_0, f_1, \cdots, f_m, h_1, \cdots, h_p$ are differentiable at the optimal point

Thus we have

$$h_i\left(\mathbf{x}^*\right) = 0, \qquad i = 1, \cdots, p$$

$$f_i\left(\mathbf{x}^*\right) \le 0, \qquad i = 1, \cdots, m$$

$$\lambda_i^* \ge 0, \qquad i = 1, \cdots, m$$

$$\lambda_i^* \cdot f_i\left(\mathbf{x}^*\right) = 0, \; i = 1, \cdots, m$$

$$\nabla f_0\left(\mathbf{x}^*\right) + \sum_{\mathrm{i}=1}^{m} \lambda_{\mathrm{i}}^* \nabla f_i\left(\mathbf{x}^*\right) + \sum_{\mathrm{i}=1}^{p} \nu_{\mathrm{i}}^* \nabla h_i\left(\mathbf{x}^*\right) = 0$$

These are known as **KKT** conditions