Descent Function

- Property
  - Its value at the optimum point for the optimization problem must be the same as that for the original cost function
  - If it is required to reduce, progress will be made towards the minimum point for the original problem
- Pshenichny's descent function (exact penalty function)

$$\Phi\left(\mathbf{x}^{(k)}\right) = f\left(\mathbf{x}^{(k)}\right) + RV\left(\mathbf{x}^{(k)}\right)$$

$$R \geq \left[r_k = \sum_{i=1}^{p} \left| v_i^{(k)} \right| + \sum_{i=1}^{m} u_i^{(k)} \right] > 0: \text{ penalty parameter}$$

$$R \geq r_k: \text{ necessary condition for convergence of the algorithm}$$

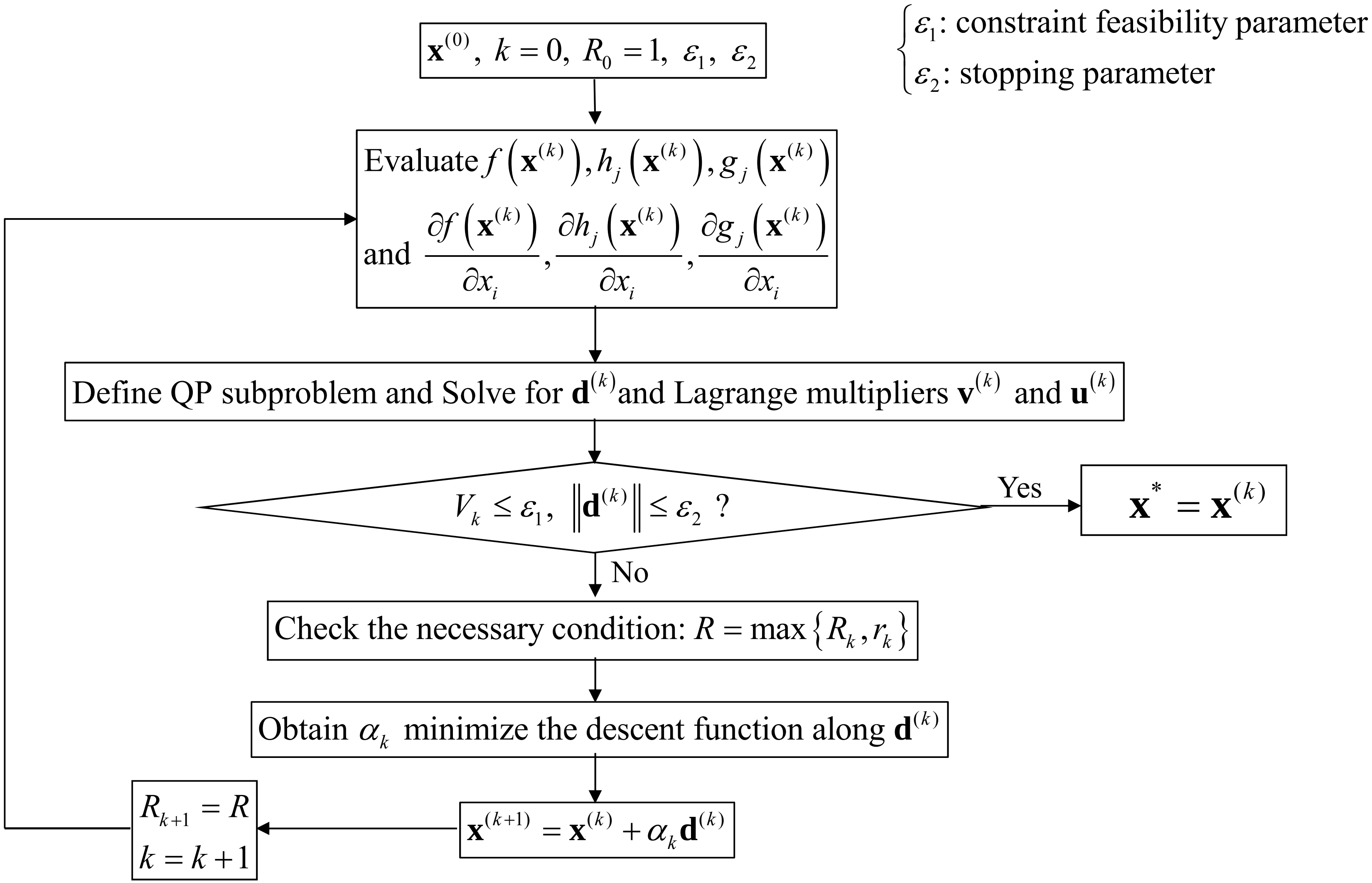
$$V\left(\mathbf{x}^{(k)}\right) = \max\left\{0; \left|h_1\right|, \ldots, \left|h_p\right|; g_1, \ldots, g_m\right\} \geq 0: \text{ maximum constraint violation}$$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)} \rightarrow \text{step size: } \Phi\left(\alpha_k\right) = \Phi\left(\mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}\right)$$

Example 12.7+8

$$\left.\begin{aligned} &\textit{Minimize } f(\boldsymbol{x}) = x_1^{\ 2} + 320x_1x_2 \\ &\textit{subject to } g_1(\boldsymbol{x}) = \frac{1}{100}(x_1 - 60x_2) \le 0 \\ &\qquad g_2(\boldsymbol{x}) = 1 - \frac{x_1(x_1 - x_2)}{3600} \le 0 \\ &\qquad g_3(\boldsymbol{x}) = -x_1 \le 0 \\ &\qquad g_4(\boldsymbol{x}) = -x_2 \le 0 \end{aligned}\right\} \rightarrow \left\{\begin{aligned} &\text{starting point } \boldsymbol{x}^{(0)} = (40, 0.5) \\ &\text{search dirction: QP subproblem} \\ &\text{initial bracketing of the step size: golden section search using } \delta\text{=0.1} \\ &\text{descent function values at these two points} : R_0 = 1 \end{aligned}\right.$$

Constrained Steepest Descent (CSD) Method



CSD Algorithm: Some Observations

- search direction
  - modification of the steepest–descent direction to satisfy constraints
  - direction obtained by projecting the steepest–descent direction on to the constraint tangent hyperplane
- observations
  - *first-order* method that can treat equality and inequality constraints
  - converges to a local minimum point starting from an arbitrary point, which is feasible or infeasible
  - potential constraint strategy in not introduced
  - Golden section search (inefficient) → Inexact line search
  - rate of convergence: improved by including second-order information in the QP subproblem
  - starting point can affect performance of the algorithm

Practical Applications

- potential constraint strategy to define the QP subproblem
- inexact line search
- constrained quasi-Newton methods

Potential Constraint Strategy

- numerical algorithms that use gradients of only a subset of the constraints in the definition of this subproblem
  - only a subset of the inequality constraints is active at the minimum point
  - this subset of active constraints is not known a priori and must be determined as part of the solution to the problem
  - efficiency of the entire iterative process: number of gradient evaluations, dimension of the subproblem
- potential constraint index

$$I_k = \left[\left\{ j \middle| j = 1 \text{ to } p \text{ for equalities} \right\} \text{and } \left\{ i \middle| g_i\left(\boldsymbol{x}^{(k)}\right) + \varepsilon \geq 0, i = 1 \text{ to } m \right\} \right]$$

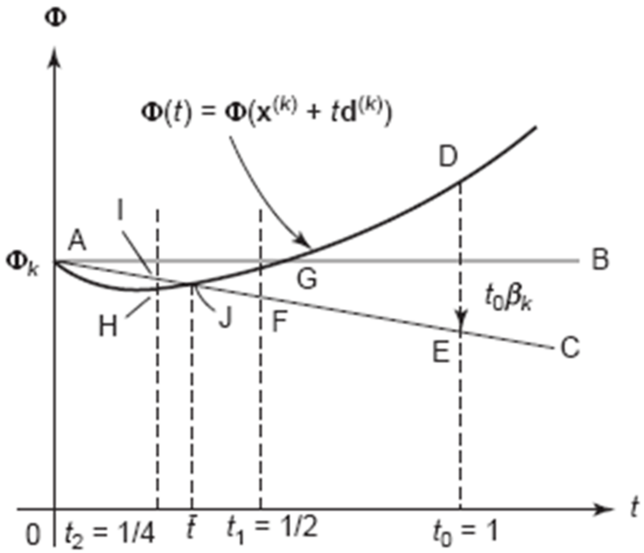
Example 13.2

$$\left.\begin{aligned}
&\textit{Minimize}\ \ f(\boldsymbol{x}) = x_1^2 - 3x_1x_2 + 4.5x_2^2 - 10x_1 - 6x_2 \\
&\textit{subject to}\ \ g_1(\boldsymbol{x}) = x_1 - x_2 \le 3 \\
&\qquad\qquad\quad g_2(\boldsymbol{x}) = x_1 + 2x_2 \le 12 \\
&\qquad\qquad\quad g_3(\boldsymbol{x}) = -x_1 \le 0 \\
&\qquad\qquad\quad g_4(\boldsymbol{x}) = -x_2 \le 0
\end{aligned}\right\}$$

$$\rightarrow \begin{cases} \text{starting point } \boldsymbol{x}^{(0)} = (4, 4) \\ \varepsilon = 0.1 \\ \text{calculate the search directions with and without the potential set strategy} \end{cases}$$

Inexact Step Size Calculation

- Step size determination
  - Inaccurate line search (Armijo's rule)
  - Trial step size
  - Parameter γ



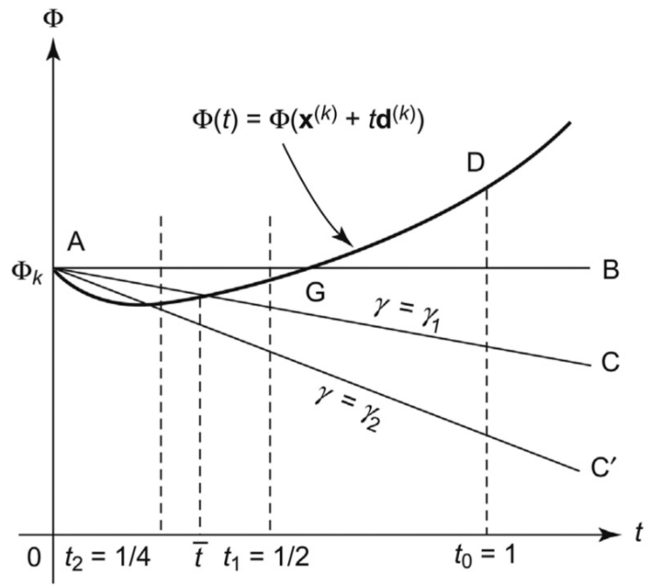
$$\Phi_k = f_k + RV_k$$

$$V_k = \max\left\{0; \left|h_1\right|, \ldots, \left|h_p\right|; g_1, \ldots, g_m\right\}$$

$$\mathbf{x}^{(k+1,j)} = \mathbf{x}^{(k)} + t_j \mathbf{d}^{(k)} \text{ where } \quad t_j = \left(\frac{1}{2}\right)^j \quad j = 0, 1, \ldots$$

$$\left[\text{descent condition}\right]$$

$$\Phi_{k+1,j} \le \Phi_k - t_j \beta_k \text{ where } \quad \beta_k = \gamma \left\|\mathbf{d}^{(k)}\right\|^2, \; 0 \le \gamma \le 1$$

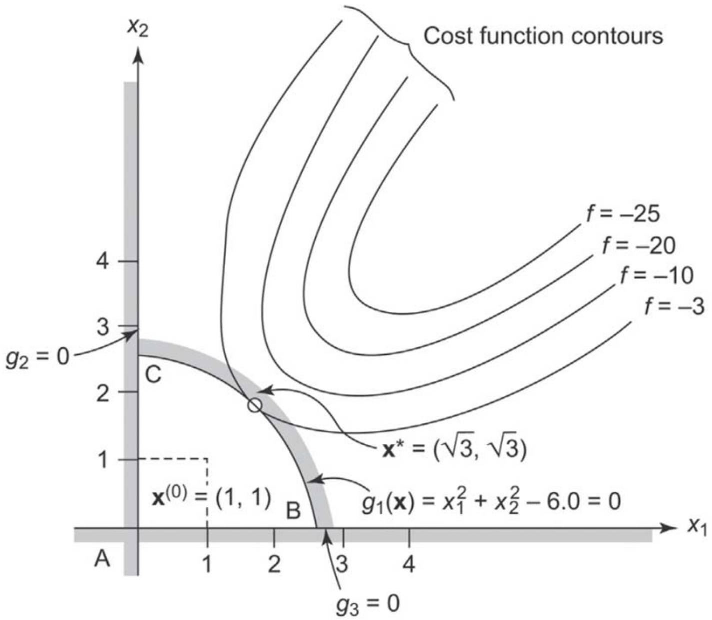


Example 13.3 ← 12.8

$$\left.\begin{aligned}&\textit{Minimize}\ f(\boldsymbol{x}) = x_1^2 + 320x_1x_2\\&\textit{subject to}\ g_1(\boldsymbol{x}) = \frac{1}{100}(x_1 - 60x_2) \le 0\\&\qquad g_2(\boldsymbol{x}) = 1 - \frac{x_1(x_1 - x_2)}{3600} \le 0\\&\qquad g_3(\boldsymbol{x}) = -x_1 \le 0\\&\qquad g_4(\boldsymbol{x}) = -x_2 \le 0\end{aligned}\right\} \rightarrow \begin{cases}\text{starting point: } \boldsymbol{x}^{(0)} = (40, 0.5)\\\text{search dirction: } \boldsymbol{d}^{(0)} = (25.6, 0.45)\\\text{Lagrange multipliers: } \boldsymbol{u} = \begin{bmatrix}16300 & 19400 & 0 & 0\end{bmatrix}^T\\\gamma = 0.5\\\text{calculate the step size using the inexact line search}\end{cases}$$

Example 13.4 ← 12.4 ← 4.31

$$\left.\begin{aligned} &\textit{Minimize}\ \ f(\boldsymbol{x}) = x_1^{\,2} + x_2^{\,2} - 3x_1x_2 \\ &\textit{subject to}\ \ g_1(\boldsymbol{x}) = \frac{1}{6}x_1^{\,2} + \frac{1}{6}x_2^{\,2} - 1 \le 0 \\ &\qquad\qquad g_2(\boldsymbol{x}) = -x_1 \le 0 \\ &\qquad\qquad g_3(\boldsymbol{x}) = -x_2 \le 0 \end{aligned}\right\} \rightarrow \left\{\begin{aligned} &x^{(0)} = (1,1) \\ &R_0 = 10 \\ &\gamma = 0.5 \\ &\varepsilon_1 = \varepsilon_2 = 0.001 \\ &\text{Perform only two iterations of CSD method} \end{aligned}\right.$$



Sequential Quadratic Programming (SQP)

- QP subproblem $\leftarrow$ curvature information of Lagrange function into the quadratic cost function
  - Constrained Quasi-Newton Methods
  - Constrained Variable Metric(CVM)
  - Recursive Quadratic Programming(RQP)
- Gradient of the Lagrange function at the two points $\rightarrow$ Approximate Hessian of the Lagrange function
- quite simple and straightforward, but very effective

Derivation of QP subproblem

$$\left.\begin{array}{l}\text{minimize } \; f(\boldsymbol{x}) \\ \text{subject to } \; h_i(\boldsymbol{x}) = 0; \quad i = 1, \ldots, p\end{array}\right\} \rightarrow L(\boldsymbol{x}, \boldsymbol{v}) = f(\boldsymbol{x}) + \sum_{i=1}^{p} v_i h_i(\boldsymbol{x}) = f(\boldsymbol{x}) + \boldsymbol{v} \cdot \boldsymbol{h}(\boldsymbol{x})$$

$$(\text{KKT necessary conditions})$$

$$\left.\begin{array}{l}\nabla L(\boldsymbol{x}, \boldsymbol{v}) = \nabla f(\boldsymbol{x}) + \boldsymbol{v} \cdot \nabla \boldsymbol{h}(\boldsymbol{x}) = 0 \\ h_i(\boldsymbol{x}) = 0; \quad i = 1, \ldots, p\end{array}\right\} \rightarrow \boldsymbol{F}(\boldsymbol{y}) = 0 \quad \text{where} \quad \boldsymbol{F} = \begin{bmatrix} \nabla L \\ \boldsymbol{h} \end{bmatrix}_{(n+p)\times 1}, \boldsymbol{y} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{v} \end{bmatrix}_{(n+p)\times 1}$$

$$\boldsymbol{y}^{(k)} \rightarrow \Delta \boldsymbol{y}^{(k)}? : \text{ linear Taylor's expansion} \rightarrow \nabla \boldsymbol{F}^T\left(\boldsymbol{y}^{(k)}\right) \Delta \boldsymbol{y}^{(k)} = -\boldsymbol{F}\left(\boldsymbol{y}^{(k)}\right)$$

$$\begin{bmatrix} \nabla^2 L & \boldsymbol{N} \\ \boldsymbol{N}^T & \boldsymbol{0} \end{bmatrix}^{(k)} \begin{bmatrix} \Delta \boldsymbol{x} \\ \Delta \boldsymbol{v} \end{bmatrix}^{(k)} = -\begin{bmatrix} \nabla L \\ \boldsymbol{h} \end{bmatrix}^{(k)} \rightarrow \begin{cases} \nabla^2 L^{(k)} \Delta \boldsymbol{x}^{(k)} + \boldsymbol{N}^{(k)} \underbrace{\Delta \boldsymbol{v}^{(k)}}_{\boldsymbol{v}^{(k+1)} - \boldsymbol{v}^{(k)}} = -\underbrace{\nabla L^{(k)}}_{\nabla f\left(\boldsymbol{x}^{(k)}\right) + \boldsymbol{N}^{(k)} \boldsymbol{v}^{(k)}} \\ \rightarrow \nabla^2 L^{(k)} \Delta \boldsymbol{x}^{(k)} + \boldsymbol{N}^{(k)} \boldsymbol{v}^{(k+1)} = -\nabla f\left(\boldsymbol{x}^{(k)}\right) \end{cases}$$

$$\rightarrow \begin{bmatrix} \nabla^2 L & \boldsymbol{N} \\ \boldsymbol{N}^T & \boldsymbol{0} \end{bmatrix}^{(k)} \begin{bmatrix} \Delta \boldsymbol{x}^{(k)} \\ \boldsymbol{v}^{(k+1)} \end{bmatrix} = -\begin{bmatrix} \nabla f \\ \boldsymbol{h} \end{bmatrix}^{(k)}$$

$$\Leftrightarrow (\text{QP subproblem}) \begin{cases} \text{minimize } \; \nabla f\left(\boldsymbol{x}^{(k)}\right)^T \Delta \boldsymbol{x}^{(k)} + \dfrac{1}{2} \Delta \boldsymbol{x}^{(k)T} \nabla^2 L \Delta \boldsymbol{x}^{(k)} \\ \text{subject to } \; h_i\left(\boldsymbol{x}^{(k)}\right) + n_i\left(\boldsymbol{x}^{(k)}\right)^T \Delta \boldsymbol{x}^{(k)} = 0; \quad i = 1, \ldots, p \end{cases}$$

Quasi-Newton Hessian Approximation

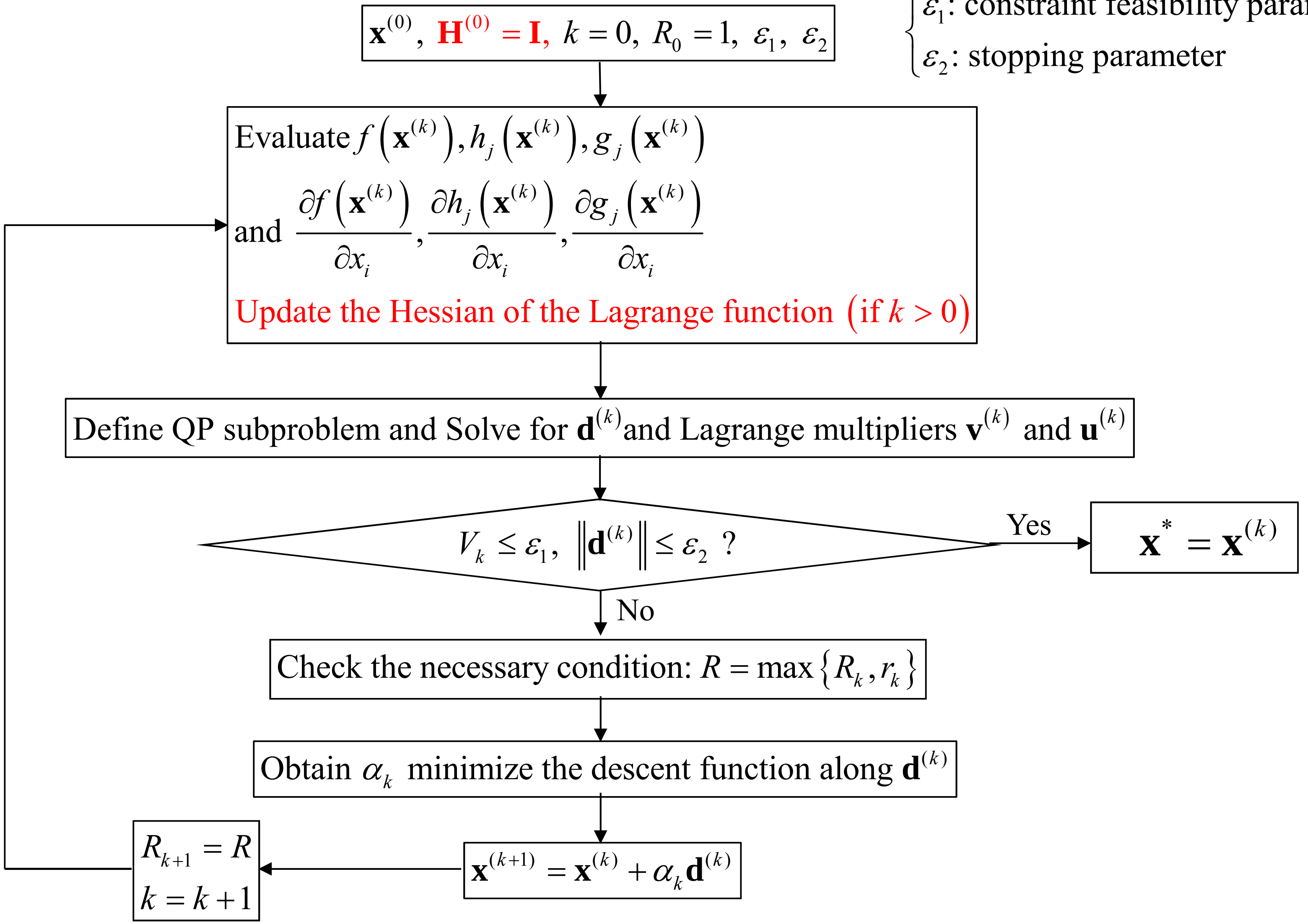
$$\left.\begin{aligned}
&\mathbf{s}^{(k)} = \alpha_k \mathbf{d}^{(k)} \\
&\mathbf{z}^{(k)} = \mathbf{H}^{(k)}\mathbf{s}^{(k)} \\
&\mathbf{y}^{(k)} = \nabla L\left(\mathbf{x}^{(k+1)}, \mathbf{u}^{(k)}, \mathbf{v}^{(k)}\right) - \nabla L\left(\mathbf{x}^{(k)}, \mathbf{u}^{(k)}, \mathbf{v}^{(k)}\right) \\
&\xi_1 = \mathbf{s}^{(k)} \cdot \mathbf{y}^{(k)} \\
&\xi_2 = \mathbf{s}^{(k)} \cdot \mathbf{z}^{(k)} \\
&\theta = \begin{cases} 1 & \text{if } \xi_1 \ge 0.2\xi_2 \\ \dfrac{0.8\xi_2}{\xi_2 - \xi_1} & \text{otherwise} \end{cases} \\
&\mathbf{w}^{(k)} = \theta \mathbf{y}^{(k)} + (1-\theta)\mathbf{z}^{(k)} \\
&\xi_3 = \mathbf{s}^{(k)} \cdot \mathbf{w}^{(k)}
\end{aligned}\right\} \rightarrow \mathbf{H}^{(k+1)} = \mathbf{H}^{(k)} + \frac{\mathbf{w}^{(k)} \cdot \mathbf{w}^{(k)T}}{\xi_3} - \frac{\mathbf{z}^{(k)} \cdot \mathbf{z}^{(k)T}}{\xi_2}$$

[BFGS]

$$H^{(k+1)} = H^{(k)} + \frac{y^{(k)}y^{(k)T}}{y^{(k)T}s^{(k)}} - \frac{H^{(k)}s^{(k)}s^{(k)T}H^{(k)}}{s^{(k)T}H^{(k)}s^{(k)}} \xrightarrow[]{\substack{s^{(k)}=x^{(k+1)}-x^{(k)}=\alpha_k d^{(k)} \\ H^{(k)}s^{(k)}=-\alpha_k c^{(k)}}} H^{(k+1)} = H^{(k)} + \frac{y^{(k)}y^{(k)T}}{y^{(k)T}s^{(k)}} + \frac{c^{(k)}c^{(k)T}}{c^{(k)T}d^{(k)}}$$

SQP Algorithm

ε_1: constraint feasibility parameter
ε_2: stopping parameter



Examples

$$
\begin{aligned}
&\textit{Minimize } f(\boldsymbol{x}) = x_1^{\ 2} + x_2^{\ 2} - 3x_1x_2 \\
&\textit{subject to } g_1(\boldsymbol{x}) = \frac{1}{6}x_1^{\ 2} + \frac{1}{6}x_2^{\ 2} - 1 \le 0 \\
&\qquad\qquad\quad g_2(\boldsymbol{x}) = -x_1 \le 0 \\
&\qquad\qquad\quad g_3(\boldsymbol{x}) = -x_2 \le 0
\end{aligned}
$$

- Example 4.31: KKT conditions
- Example 12.4: SLP
- Example 13.4: CSD
- Example 13.9: SQP

Descent Functions in SQP

- play an important role in QP methods
- Nondifferentiable vs. differentiable

$$\Phi_P = f\left(\mathbf{x}^{(k)}\right) + R\left[\max\left\{0; |h_1|, \ldots, |h_p|; g_1, \ldots, g_m\right\}\right]$$

$$\Phi_H = f\left(\mathbf{x}^{(k)}\right) + \sum_{i=1}^{p} r_i^{(k)} \left|h_i^{(k)}\right| + \sum_{i=1}^{m} \mu_i^{(k)} \max\left\{0, g_i\right\}$$

$$\Phi_A = f(\mathbf{x}) + P_1(\mathbf{v},\mathbf{h}) + P_2(\mathbf{u},\mathbf{h}) \quad \text{where} \quad \begin{cases} P_1(\mathbf{v},\mathbf{h}) = \sum_{i=1}^{p} \left(v_i h_i + \frac{1}{2} r_i h_i^2 \right) \\ P_2(\mathbf{u},\mathbf{h}) = \sum_{i=1}^{m} \begin{cases} \left(u_i g_i + \frac{1}{2} \mu_i g_i^2 \right) & \text{if } \left(g_i + \frac{u_i}{\mu_i} \right) \geq 0 \\ -\frac{1}{2} \frac{u_i^2}{\mu_i} & \text{otherwise} \end{cases} \end{cases}$$

Other Numerical Optimization Methods

- Method of Feasible Directions
- Gradient Projection Method
- Generalized Reduced Gradient Method

Method of Feasible Directions

- Zoutendijk(1960)
- Inequality constrained problem
  - Feasible region has an 'interior'
  - Equality constraints? penalty function

$$\begin{cases} Minimize & f(\boldsymbol{x}) \\ subject\ to & g_i(\boldsymbol{x}) \le 0 \quad i = 1, \ldots, m \end{cases}$$

$$I = \left\{ j : g_j\left(\boldsymbol{x}^{(k)}\right) + \varepsilon \ge 0,\ j = 1, \ldots, m \right\}$$

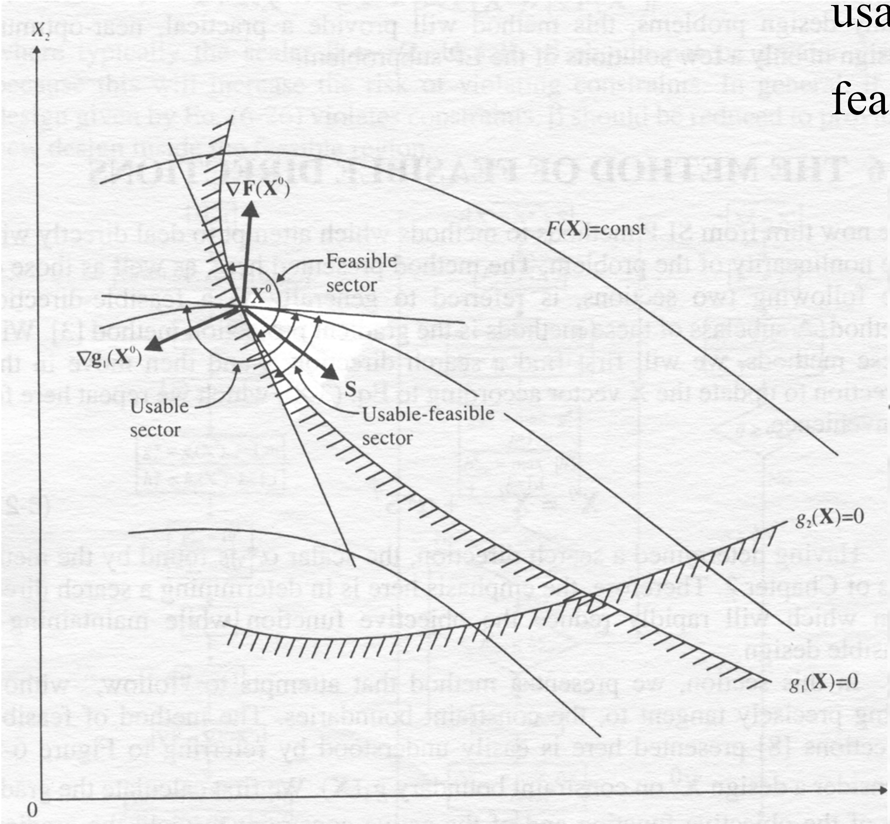
$$\alpha = \max\left\{ \nabla f^T \boldsymbol{d}, \nabla g_j{}^T \boldsymbol{d} \text{ for each } j \in I \right\}$$

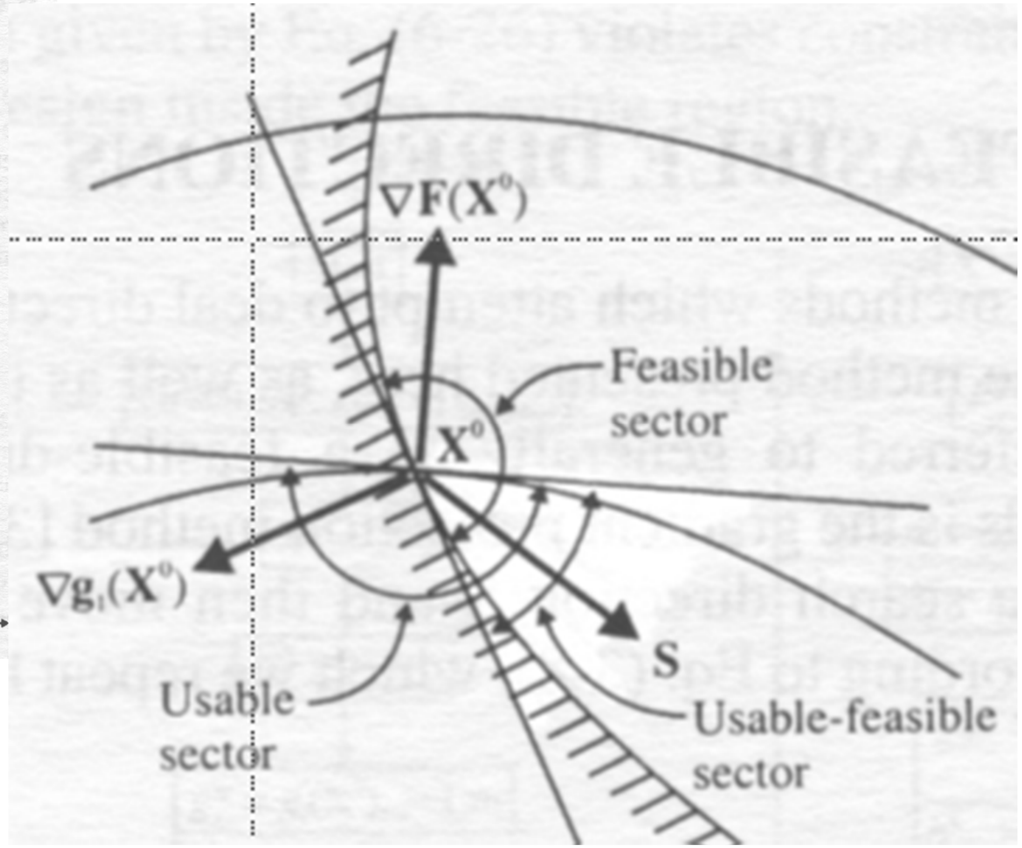
- Direction
  - Define a linearized subproblem @ the current feasible point
- Step size
  - Reduce the cost function as well as maintain feasibility

Usable-Feasible Direction

$$\text{usable direction}: \left[\nabla f\left(\boldsymbol{x}^{(k)}\right)^T \boldsymbol{d} \le 0\right], f\left(\boldsymbol{x}^{(k+1)}\right) \le f\left(\boldsymbol{x}^{(k)}\right)$$

$$\text{feasible direction}: \left[\nabla g\left(\boldsymbol{x}^{(k)}\right)^T \boldsymbol{d} \le 0\right]$$





FDM: Direction-finding

$$
\left.\begin{aligned}
&\textit{Minimize} \quad \alpha \\
&\textit{subject to} \quad \nabla f^T \boldsymbol{d} \le \alpha \\
&\qquad \nabla g_j{}^T \boldsymbol{d} \le \alpha \quad \text{for each } j \in I \\
&\xrightarrow{\text{push-off factor}} \nabla g_j{}^T \boldsymbol{d} \le \theta_j \alpha \\
&\qquad -1 \le d_i \le 1 \quad i = 1, \ldots, n
\end{aligned}\right\}
\xrightarrow{s_i = d_i + 1, \beta = -\alpha}
\begin{cases}
\textit{Minimize} \quad -\beta \\
\textit{subject to} \quad \nabla f^T \boldsymbol{s} + \beta \le c_0 \\
\qquad \nabla g_j{}^T \boldsymbol{s} + \theta_j \beta \le c_j \quad \text{for each } j \in I \\
\qquad s_i \le 2 \quad i = 1, \ldots, n \\
\qquad s_i \ge 0, \beta \ge 0 \\
\text{where} \quad c_0 = \sum_{i=1}^{n} \partial f / \partial x_i, c_j = \sum_{i=1}^{n} \partial g_j / \partial x_i
\end{cases}
$$

$$
\xrightarrow{s_i = d_i + 1, \beta = -\alpha}
\begin{cases}
\textit{Minimize} \quad -\beta \\
\textit{subject to} \quad \nabla f^T \boldsymbol{s} + \beta \le c_0 \\
\qquad \nabla g_j{}^T \boldsymbol{s} + \theta_j \beta \le c_j \quad \text{for each } j \in I \\
\qquad s_i \le 2 \quad i = 1, \ldots, n \\
\qquad s_i \ge 0, \beta \ge 0 \\
\text{where} \quad c_0 = \sum_{i=1}^{n} \partial f / \partial x_i, c_j = \sum_{i=1}^{n} \partial g_j / \partial x_i
\end{cases}
\Leftrightarrow
\begin{cases}
\textit{Minimize} \quad \boldsymbol{c}^T \boldsymbol{x} \\
\textit{subject to} \quad A\boldsymbol{x} \le b \\
\qquad \boldsymbol{x} \ge 0 \\
\text{where} \quad \boldsymbol{x}^T = \left(\boldsymbol{s}^T, \beta\right)
\end{cases}
$$

Push-off factor

- Nonlinear and convex constraints
  - Small move $\rightarrow$constraints violation

$$\nabla g\left(\boldsymbol{x}^{(k)}\right)^T \boldsymbol{d} \le \boldsymbol{\theta}\alpha$$

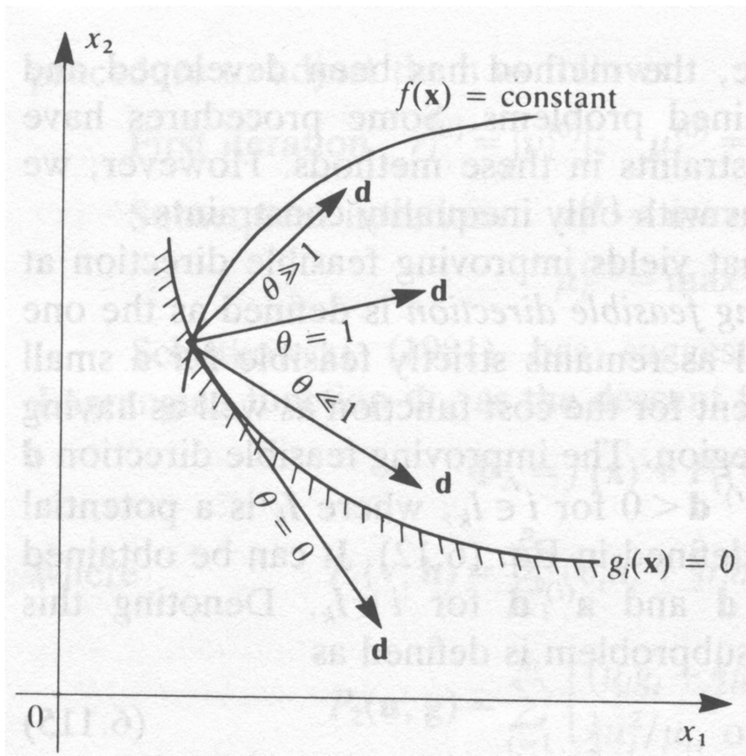
($\boldsymbol{\theta}$: nonnegative constant)

$$\theta_j = \begin{cases} 0 & \text{for linear constraints} \\ \left[1 - \dfrac{g_j(\boldsymbol{x})}{\varepsilon}\right]^2 \theta_0 & (\text{usually } \theta_0 = 1.0) \end{cases}$$

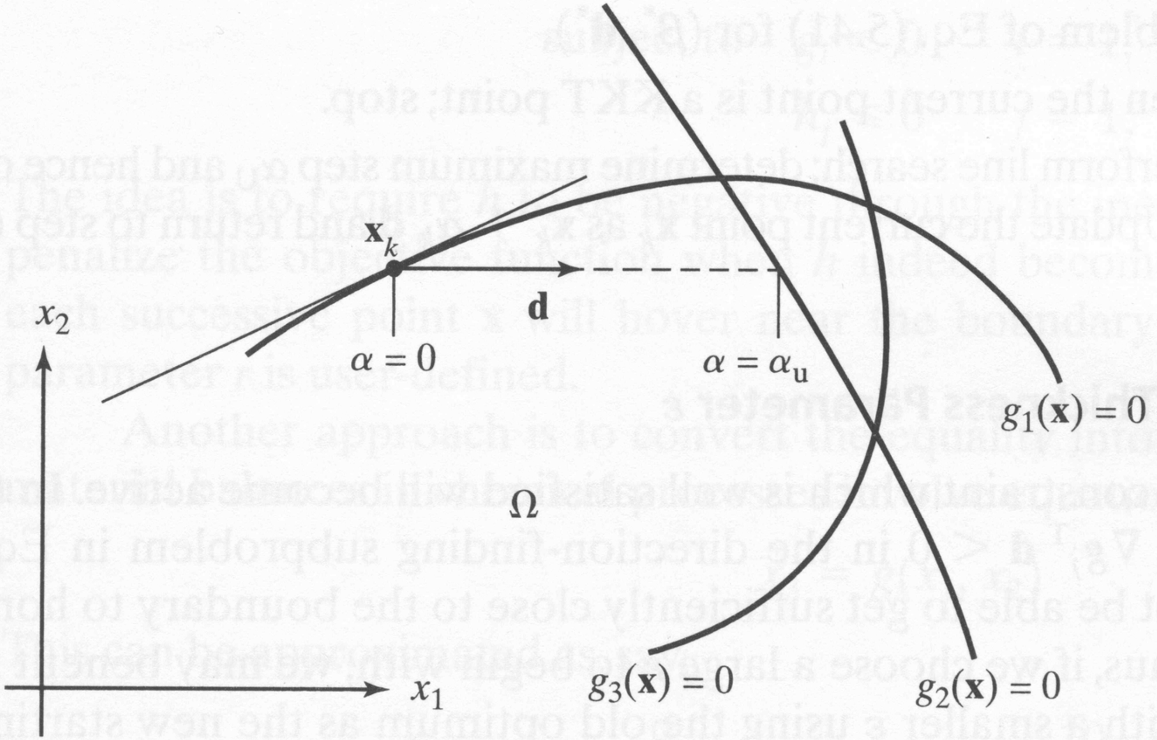
pushing harder

as the constraint becomes more critical

$$\theta_j = 0\left(g_j(\boldsymbol{x}) = \varepsilon\right) \rightarrow \theta_j = 1\left(g_j(\boldsymbol{x}) = 0\right)$$



FDM: Line Search



$$f(\alpha) \equiv f(\mathbf{x}_k + \alpha \mathbf{d}), \quad f'(\bar{\alpha}) \equiv \left.\frac{df}{d\alpha}\right|_{\alpha = \bar{\alpha}}$$

If $f'(\alpha_u) \le 0$ Then $\alpha^* = \alpha_u$

If $f'(\alpha_u) > 0$ Then minimize $f(\alpha)$ to get α^*, $0 < \alpha < \alpha_u$

$$\textit{Minimize} \quad f(\alpha) \equiv f\left(\boldsymbol{x}^{(k)} + \alpha \boldsymbol{d}\right)$$
$$\textit{subject to} \quad g_i(\alpha) \equiv g_i\left(\boldsymbol{x}^{(k)} + \alpha \boldsymbol{d}\right) \le 0$$
$$i = 1, \ldots, m$$

$\rightarrow$ Set α_{U1} (maximum step based on bounds only)

from $x_i^{\,L} \le x_i^{(k)} + \alpha d_i \le x_i^{\,U}$

$$\rightarrow \begin{cases} -g_{low} \le g_{\max} = \max\{g_j\} \le 0 \\ \rightarrow \alpha_U = \alpha_{U1} \\ \textit{otherwise}, \text{find } \alpha_U, \ (0 < \alpha_U < \alpha_{U1}) \\ \text{to the nearest intersecting} \\ \text{constraint : zero - finding} \end{cases}$$

Major Shortcomings

- For general problems, there may not exist any feasible direction.
  - Relax definition of feasibility or allow points to deviate
  - Introduce concept of moving along curves rather than straight lines
- Feasible direction methods can be subject to jamming/zigzagging, that is, it does not converge to a constrained local minimum.
  - In Zoutendijk’s method, the method for finding a feasible direction changes is another constraint

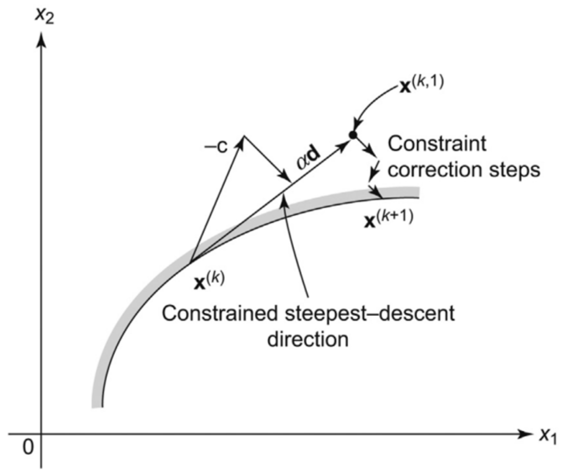
Gradient Projection Methods

- Rosen(1960)
  - Does not require the solution of an auxiliary optimization problem to find the usable feasible direction
  - Effective for linear constraints
- Determine the direction by projecting the steepest descent direction onto the tangent plane
- Major task is to calculate projection matrix ***P*** and subsequent feasible direction vector ***d***.

$$\text{minimize} \quad f(\boldsymbol{x})$$

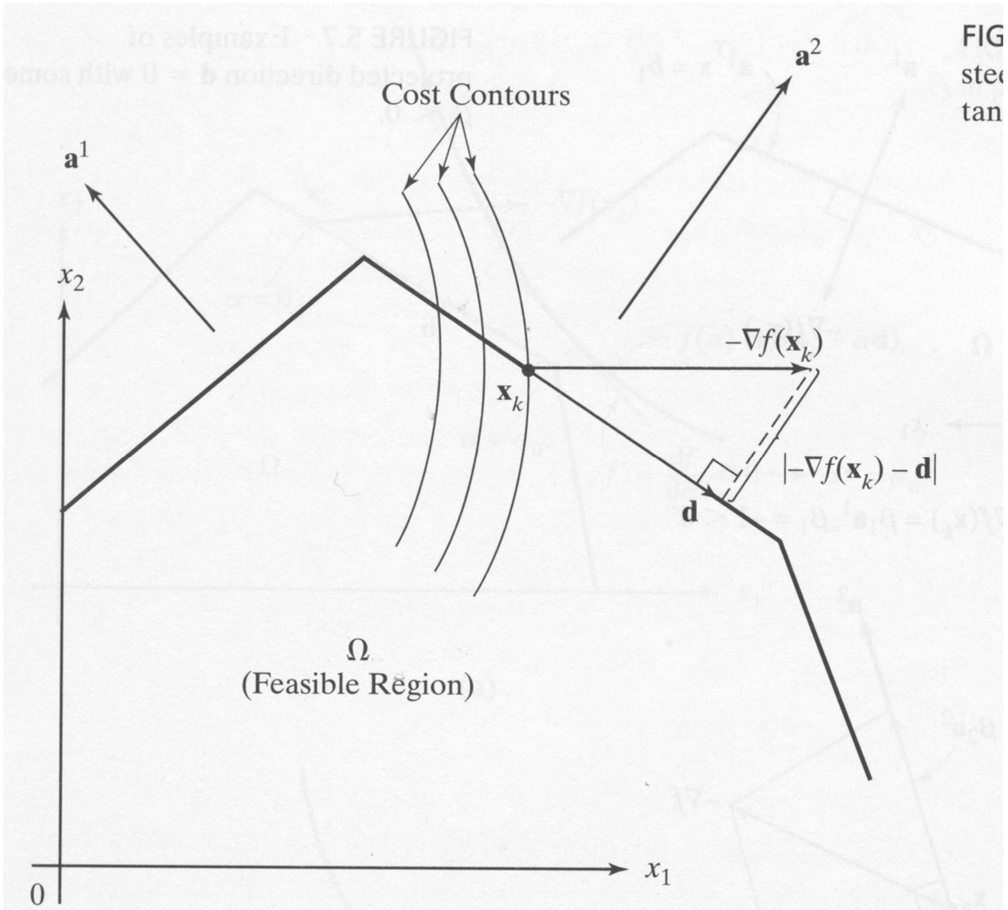
$$\text{subject to} \quad g_j(\boldsymbol{x}) = \sum_{i=1}^{n} a_{ij}x_i - b_j \le 0, \quad j = 1, \ldots, m$$

$$\underbrace{\boldsymbol{N}}_{n\times J} = [\nabla g_1 \quad \ldots \quad \nabla g_J]$$



Gradient Projection: Direction

- ***P*** projects the vector $-\nabla f(\boldsymbol{x})$ onto the intersection of all the hyperplanes perpendicular to the vectors $\nabla g_j(\boldsymbol{x})$, $j \in J$



Direction-finding

$$\begin{cases} \text{minimize } \left[-\nabla f(\boldsymbol{x}) - \boldsymbol{d}\right]^T \left[-\nabla f(\boldsymbol{x}) - \boldsymbol{d}\right] \\ \text{subject to } \boldsymbol{N}^T \boldsymbol{d} = 0 \end{cases}$$

$$\rightarrow L(\boldsymbol{d}, \beta) = \left[\nabla f(\boldsymbol{x}) + \boldsymbol{d}\right]^T \left[\nabla f(\boldsymbol{x}) + \boldsymbol{d}\right] + \beta^T \boldsymbol{N}^T \boldsymbol{d}$$

$$\begin{cases} \dfrac{\partial L}{\partial \boldsymbol{d}} = \nabla f(\boldsymbol{x}) + \boldsymbol{d} + N\beta = \boldsymbol{0} \rightarrow \boldsymbol{d} = -\nabla f(\boldsymbol{x}) - N\beta \\ \dfrac{\partial L}{\partial \beta} = \boldsymbol{N}^T \boldsymbol{d} = \boldsymbol{0} \rightarrow -\left(\boldsymbol{N}^T \nabla f(\boldsymbol{x}) + \boldsymbol{N}^T \boldsymbol{N}\beta\right) = \boldsymbol{0} \rightarrow \beta = -\left(\boldsymbol{N}^T \boldsymbol{N}\right)^{-1} \boldsymbol{N}^T \nabla f(\boldsymbol{x}) \end{cases}$$

$$\boldsymbol{d} = -\nabla f(\boldsymbol{x}) + \boldsymbol{N}\left(\boldsymbol{N}^T \boldsymbol{N}\right)^{-1} \boldsymbol{N}^T \nabla f(\boldsymbol{x}) = \left[I - \boldsymbol{N}\left(\boldsymbol{N}^T \boldsymbol{N}\right)^{-1} \boldsymbol{N}^T\right]\left[-\nabla f(\boldsymbol{x})\right] = \boldsymbol{P}\left[-\nabla f(\boldsymbol{x})\right]$$

$$P = I - \boldsymbol{N}\left(\boldsymbol{N}^T \boldsymbol{N}\right)^{-1} \boldsymbol{N}^T \ : \text{projection matrix}$$

$$\boldsymbol{d}^{(k)} = -\frac{\boldsymbol{P}^{(k)} \nabla f\left(\boldsymbol{x}^{(k)}\right)}{\left\|\boldsymbol{P}^{(k)} \nabla f\left(\boldsymbol{x}^{(k)}\right)\right\|}$$

$$\boldsymbol{d}^{(k)} = 0 \rightarrow -\nabla f\left(\boldsymbol{x}^{(k)}\right) = N\lambda = \sum_{j=1}^{J} \lambda_j \nabla g_j \ (\text{K} - \text{T condition})$$

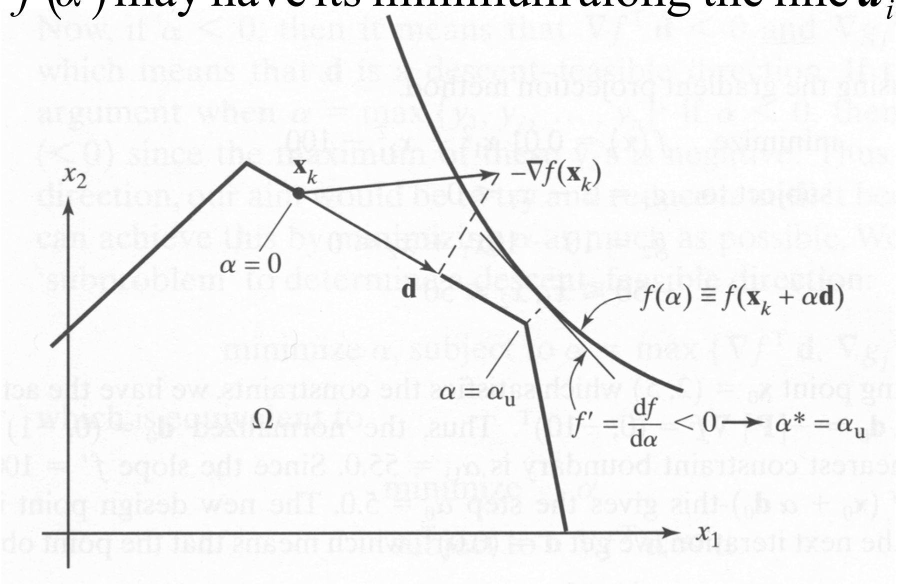
Gradient Projection: Line Search

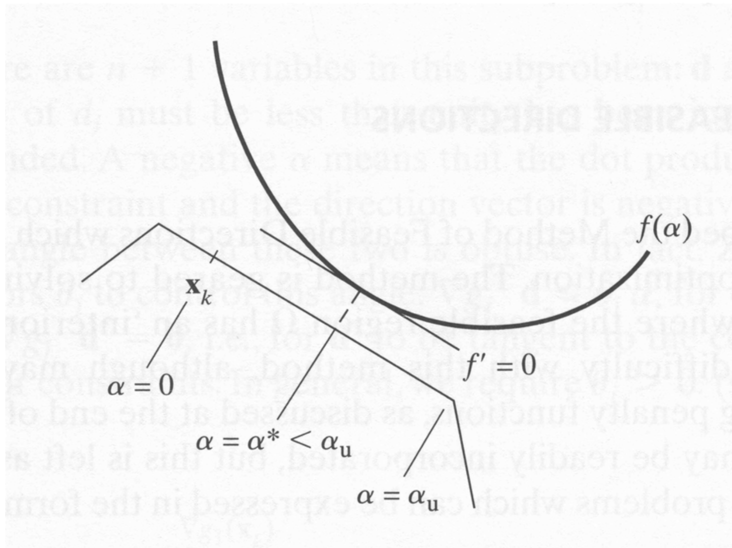
$$g_j(\alpha) = g_j\left(\boldsymbol{x}^{(k)} + \alpha \boldsymbol{d}^{(k)}\right) = \sum_{i=1}^{n} a_{ij}\left(x_i + \alpha d_i\right) - b_j$$

$$= \sum_{i=1}^{n} a_{ij} x_i - b_j + \alpha \sum_{i=1}^{n} a_{ij} d_i = g_j\left(\boldsymbol{x}^{(k)}\right) + \alpha \sum_{i=1}^{n} a_{ij} d_i, \quad j = 1, \ldots, m$$

$$\alpha_j = -\frac{g_j\left(\boldsymbol{x}^{(k)}\right)}{\sum_{i=1}^{n} a_{ij} d_i} \rightarrow \alpha_M = \min_{\alpha_j > 0}\left(\alpha_j\right) \quad (j \text{ is any integer among } 1 \textit{ to } m \text{ other than } 1,\ldots,J)$$

$f(\alpha)$ may have its minimum along the line $\boldsymbol{d}_i$ between $\alpha = 0$ and $\alpha = \alpha_M$.





Generalized Reduced Gradient Method

- Elimination of variables using the equality constraints
  – One variable can be reduced from the set x_i for each of the $m+p$ equality constraints

$$\left.\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) \\ \text{subject to} & g_i(\boldsymbol{x}) \le 0, \quad i = 1,\ldots,m \\ & h_j(\boldsymbol{x}) = 0, \quad j = 1,\ldots,l \\ & x_k^L \le x_k \le x_k^U, \quad k = 1,\ldots,n \end{array}\right\} \rightarrow \left\{\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) \\ \text{subject to} & g_i(\boldsymbol{x}) + x_{n+i} = 0, \quad i = 1,\ldots,m \\ & h_j(\boldsymbol{x}) = 0, \quad j = 1,\ldots,l \\ & x_k^L \le x_k \le x_k^U, \quad k = 1,\ldots,n \\ & x_{n+i} \ge 0, \quad i = 1,\ldots,m \end{array}\right.$$

$$\rightarrow \left\{\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) \\ \text{subject to} & \bar{h}_j(\boldsymbol{x}) = 0, \quad j = 1,\ldots,m+l \\ & x_i^L \le x_i \le x_i^U, \quad i = 1,\ldots,n+m \end{array}\right.$$

$$\boldsymbol{x} = \begin{Bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{Bmatrix}, \quad \underbrace{\boldsymbol{y} = \begin{Bmatrix} y_1 \\ \vdots \\ y_{m+l} \end{Bmatrix}}_{\text{state or dependent variables}}, \quad \underbrace{\boldsymbol{z} = \begin{Bmatrix} z_1 \\ \vdots \\ z_{n-l} \end{Bmatrix}}_{\text{design or independent variables}}$$

Reduced Gradient

$$df(\boldsymbol{x}) = \sum_{i=1}^{m+l} \frac{\partial f}{\partial y_i} dy_i + \sum_{i=1}^{n-l} \frac{\partial f}{\partial z_i} dz_i = \nabla_y^T f d\boldsymbol{y} + \nabla_z^T f d\boldsymbol{z}$$

$$d\overline{h}_i(\boldsymbol{x}) = \sum_{j=1}^{m+l} \frac{\partial \overline{h}_i}{\partial y_j} dy_j + \sum_{j=1}^{n-l} \frac{\partial \overline{h}_i}{\partial z_j} dz_j \to d\overline{\boldsymbol{h}} = \boldsymbol{B} d\boldsymbol{y} + \boldsymbol{C} d\boldsymbol{z}$$

$$\nabla_y^T f = \left\{ \begin{matrix} \dfrac{\partial f}{\partial y_1} \\ \vdots \\ \dfrac{\partial f}{\partial y_{m+l}} \end{matrix} \right\}, \nabla_z^T f = \left\{ \begin{matrix} \dfrac{\partial f}{\partial z_1} \\ \vdots \\ \dfrac{\partial f}{\partial z_{n-l}} \end{matrix} \right\}, d\boldsymbol{y} = \left\{ \begin{matrix} dy_1 \\ \vdots \\ dy_{m+l} \end{matrix} \right\}, d\boldsymbol{z} = \left\{ \begin{matrix} dz_1 \\ \vdots \\ dz_{n-l} \end{matrix} \right\}$$

$$\boldsymbol{B} = \begin{bmatrix} \dfrac{\partial \overline{h}_1}{\partial y_1} & \cdots & \dfrac{\partial \overline{h}_1}{\partial y_{m+l}} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial \overline{h}_{m+l}}{\partial y_1} & \cdots & \dfrac{\partial \overline{h}_{m+l}}{\partial y_{m+l}} \end{bmatrix}, \ \boldsymbol{C} = \begin{bmatrix} \dfrac{\partial \overline{h}_1}{\partial z_1} & \cdots & \dfrac{\partial \overline{h}_1}{\partial z_{n-l}} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial \overline{h}_{m+l}}{\partial z_1} & \cdots & \dfrac{\partial \overline{h}_{m+l}}{\partial y_{n-l}} \end{bmatrix}$$

GRG: Direction

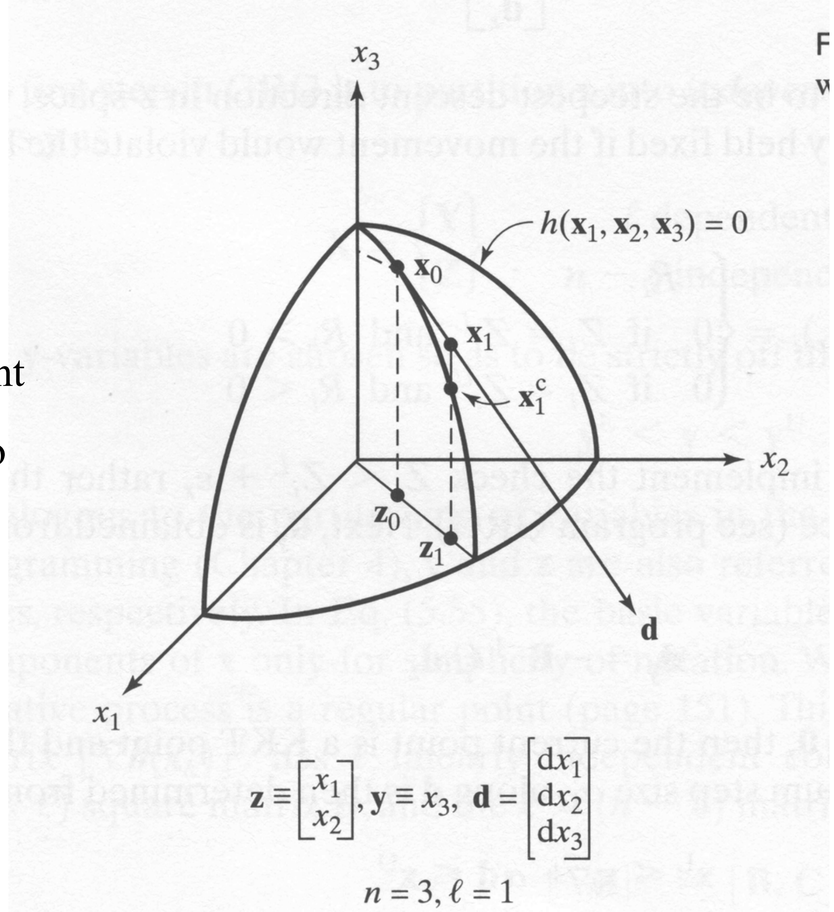
$$d\bar{\boldsymbol{h}} = \boldsymbol{B}d\boldsymbol{y} + \boldsymbol{C}d\boldsymbol{z} = 0 \ \left(\bar{\boldsymbol{h}}(\boldsymbol{x}) = 0\right) \rightarrow d\boldsymbol{y} = -\boldsymbol{B}^{-1}\boldsymbol{C}d\boldsymbol{z}$$

$$df(\boldsymbol{x}) = \left(-\nabla_y^T f \boldsymbol{B}^{-1}\boldsymbol{C} + \nabla_z^T f\right)d\boldsymbol{z} \rightarrow \frac{df(\boldsymbol{x})}{d\boldsymbol{z}} = \boldsymbol{G}_R$$

$$\boldsymbol{G}_R = \nabla_z f - \left(\boldsymbol{B}^{-1}\boldsymbol{C}\right)^T \nabla_y f \ \ : \text{ generalized reduced gradient}$$

$\rightarrow$ projection of the original n-dimensional gradient onto the $(n - m)$ dimensional feasible region described by the design variables

$$\boldsymbol{d} = \begin{bmatrix} \boldsymbol{d}_y \\ \boldsymbol{d}_z \end{bmatrix} \rightarrow \begin{cases} \boldsymbol{d}_y = -\boldsymbol{B}^{-1}\boldsymbol{C}\boldsymbol{d}_z \\ \left(\boldsymbol{d}_z\right)_i = \begin{cases} -\left(\boldsymbol{G}_R\right)_i \\ 0 \ \text{ if } \ z_i = z_i^L \ \text{ and } \ \left(\boldsymbol{G}_R\right)_i > 0 \\ 0 \ \text{ if } \ z_i = z_i^U \ \text{ and } \ \left(\boldsymbol{G}_R\right)_i < 0 \end{cases} \end{cases}$$



GRG: Line Search

$$\boldsymbol{x}^L \le \boldsymbol{x}_k + \alpha \boldsymbol{d} \le \boldsymbol{x}^U$$

$$\alpha_z = \begin{cases} \dfrac{z_i^U - z_i^{old}}{d_i} & \text{if } d_i > 0 \\ \dfrac{z_i^L - z_i^{old}}{d_i} & \text{if } d_i < 0 \end{cases}$$

$$d\boldsymbol{y} = -\boldsymbol{B}^{-1}\boldsymbol{C}d\boldsymbol{z} \xrightarrow{dz = \alpha \boldsymbol{d}} \boldsymbol{e} = -\boldsymbol{B}^{-1}\boldsymbol{C}\boldsymbol{d}$$

$$\alpha_y = \begin{cases} \dfrac{y_i^U - y_i^{old}}{e_i} & \text{if } e_i > 0 \\ \dfrac{y_i^L - y_i^{old}}{e_i} & \text{if } e_i < 0 \end{cases}$$

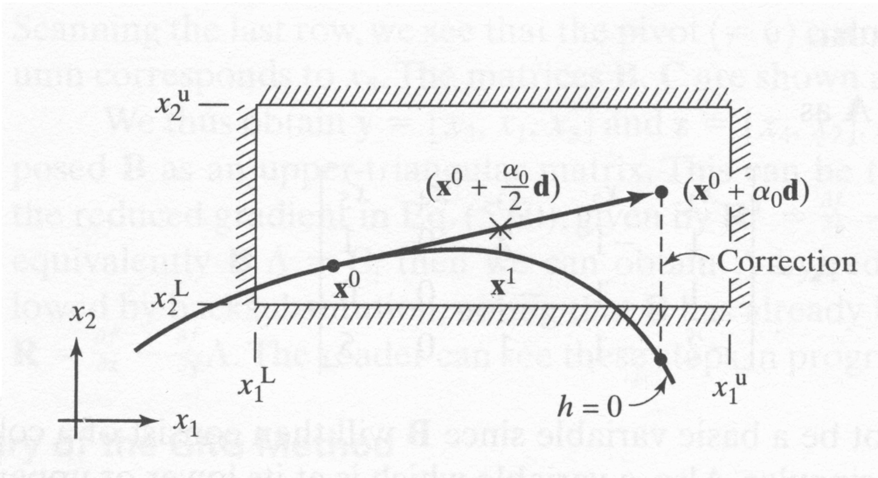
$$\alpha = \min\left(\alpha_y, \alpha_z\right)$$

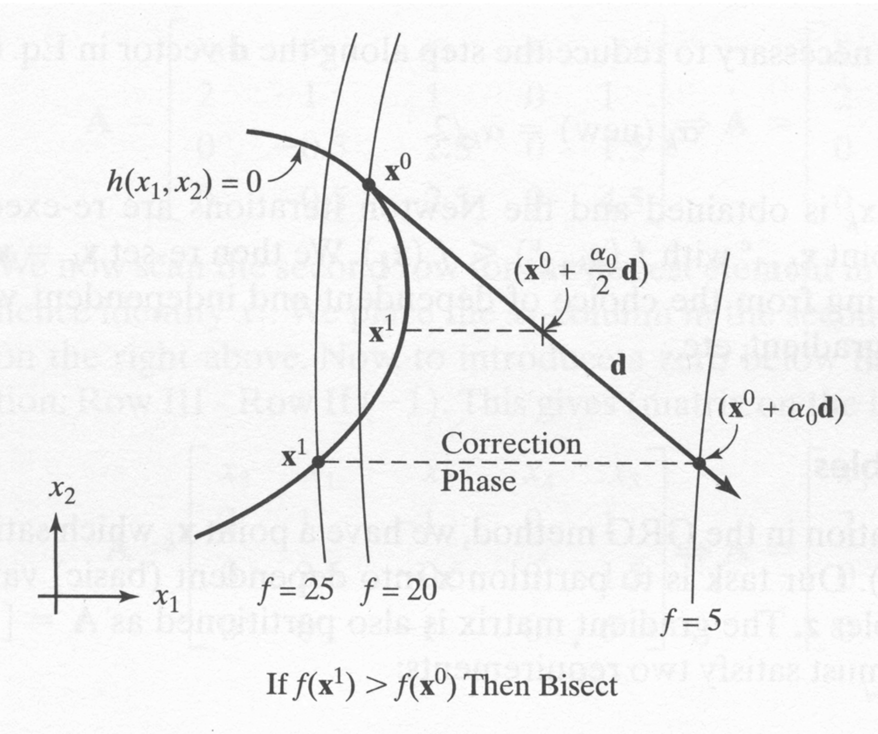
$$\boldsymbol{x}^{new} = \begin{Bmatrix} \boldsymbol{y}^{old} + d\boldsymbol{y} \\ \boldsymbol{z}^{old} + d\boldsymbol{z} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{y}^{old} + \alpha \boldsymbol{e} \\ \boldsymbol{z}^{old} + \alpha \boldsymbol{d} \end{Bmatrix}$$

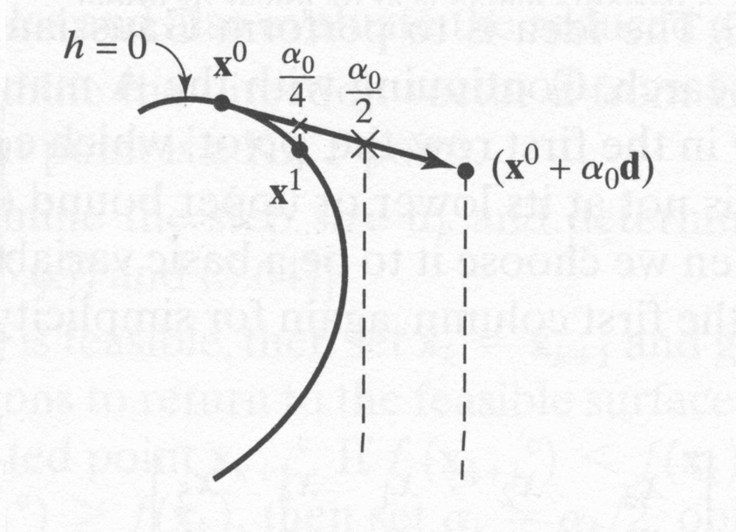
GRG: Correction

$$\overline{\boldsymbol{h}}(\boldsymbol{x})+d\overline{\boldsymbol{h}}(\boldsymbol{x})=\boldsymbol{0}\rightarrow d\boldsymbol{y}=\boldsymbol{B}^{-1}\left[-\overline{\boldsymbol{h}}(\boldsymbol{x})-\boldsymbol{C}d\boldsymbol{z}\right]\rightarrow \boldsymbol{y}^{new}=\boldsymbol{y}^{old}+d\boldsymbol{y}$$

– Increased function value
– Bound violation
– Increased constraint violation







GRG: Algorithm

- Start with an initial trial vector $\mathbf{x}_0$. Specify the design and state variables.
  - State variable: avoid singularity of the matrix, B / slack variable
  - Design variable: lower and upper bound
- Compute the generalized reduced gradient
- Test for convergence
- Determine the search direction
- Find the minimum along the search direction
  - Find an estimate for α as the distance to the nearest side constraint
  - If the vector $\mathbf{x}_{new}$ corresponding to α is found infeasible, $\mathbf{z}_{new}$ is held constant and $\mathbf{y}_{new}$ is modified

Remarks

- Closely related to simplex LP method because variables are split into basic and non-basic groups.
- From a theoretical viewpoint, the method behaves very much like the gradient projection method.
- Like gradient projection method, it can be regarded as a steepest descent method applied on the surface defined by the active constraints.
- Reduced gradient method seems to be better than gradient projection methods.

Advantages of Primal Methods

- If the process is terminated before reaching the solution (practically for nonlinear problems), the terminating point is feasible and probably near optimal.
- Guaranteed that if they generate a convergent sequence, then the limit point of that sequence must be at least a local constrained minimum.
- Do not rely on a special problem structure, such as convexity, hence applicable to general NLP.
- Competitive convergence rates, particularly for linear constraints

Disadvantages of Primal Methods

- They require a Phase I procedure to obtain an initial feasible point.
- They are all plagued, particularly for problems with nonlinear constraints, with computational difficulties arising from the necessity to remain within the feasible region as the method progresses.
- Some methods can fail to converge for problems with inequality constraints unless elaborate precautions are taken.