SIMPLEX LIKE (*aka* REDUCED GRADIENT) METHODS

The REDUCED GRADIENT algorithm and its variants such as the CONVEX SIMPLEX METHOD (CSM) and the GENERALIZED REDUCED GRADIENT (GRG) algorithm are approximation methods that are similar in spirit to the Simplex method for LP.

The original reduced gradient method and CSM operate on a linear constraint set with nonnegative decision variables, while GRG generalizes the procedure to nonlinear constraints, possibly with lower and upper bounds on decision variables.

REDUCED GRADIENT METHOD (Wolfe)

This is a generalization of the simplex method for LP, to solve problems with nonlinear objectives:

Minimize $f(\boldsymbol{x})$

st $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \qquad \boldsymbol{x} \geq 0.$

Here we assume that $\boldsymbol{A}$ is an $(m \times n)$ matrix of rank m, and that $\boldsymbol{b}, \boldsymbol{x} \in \mathbb{R}^n$.

<u>At iteration *i*</u>

We have the n decision variables partitioned as BASIC and NON-BASIC, i.e,

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_{\mathrm{B}} \\ \hdashline \boldsymbol{x}_{\mathrm{N}} \end{bmatrix}, \quad \boldsymbol{A}_{m \times n} = \left[\underset{m \times m}{\boldsymbol{B}} \;\vdots\; \underset{m \times (n-m)}{\boldsymbol{N}} \right], \qquad \mathrm{Rank}(\boldsymbol{B}) = m \implies \boldsymbol{B}^{-1} \text{ exists}$$

Thus $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \implies [\boldsymbol{B} \;\vdots\; \boldsymbol{N}] \begin{bmatrix} \boldsymbol{x}_{\mathrm{B}} \\ \hdashline \boldsymbol{x}_{\mathrm{N}} \end{bmatrix} = \boldsymbol{b} \implies \boldsymbol{B}\boldsymbol{x}_{\mathrm{B}} + \boldsymbol{N}\boldsymbol{x}_{\mathrm{N}} = \boldsymbol{b}$

(After possible reordering), let $\boldsymbol{x}^{\boldsymbol{i}} = \begin{bmatrix} \boldsymbol{x}^{\boldsymbol{i}}_{\mathrm{B}} \\ \hdashline \boldsymbol{x}^{\boldsymbol{i}}_{\mathrm{N}} \end{bmatrix}$, be the current design.

A requirement for guaranteeing convergence is that no degeneracy is permitted at any iteration, i.e., $\boldsymbol{x}_B > \mathbf{0}$. If more than m variables are positive at the iteration, then the m largest (positive) variables are selected as the basic variables.

We wish to decrease $f(\boldsymbol{x})$ while maintaining feasibility. Suppose we change the non-basic variables by an amount $\boldsymbol{\delta}_N$ from $\boldsymbol{x}_N^i$, and the basic variables by an amount $\boldsymbol{\delta}_B$ from $\boldsymbol{x}_B^i$. Then $\boldsymbol{\delta} = \begin{bmatrix}\boldsymbol{\delta}_B \\ \boldsymbol{\delta}_N\end{bmatrix}$.

As $\boldsymbol{x}_N$ changes from $\boldsymbol{x}_N^i$ by $\boldsymbol{\delta}_N$, $\boldsymbol{x}_B$ should be changed by $\boldsymbol{\delta}_B$ in such a way that the new point $\boldsymbol{x}^{i+1} = \boldsymbol{x}^i + \boldsymbol{\delta}$ is still feasible, i.e.,

$$\boldsymbol{A}\begin{bmatrix}\boldsymbol{x}_B^{i+1} \\ \boldsymbol{x}_N^{i+1}\end{bmatrix} = [\boldsymbol{B} \;\vdots\; \boldsymbol{N}]\left\{\begin{bmatrix}\boldsymbol{x}_B^i \\ \cdots \\ \boldsymbol{x}_N^i\end{bmatrix} + \begin{bmatrix}\boldsymbol{\delta}_B \\ \boldsymbol{\delta}_N\end{bmatrix}\right\} = \boldsymbol{b}$$

$$\Rightarrow \quad \boldsymbol{B}\boldsymbol{x}_B^i + \boldsymbol{B}\boldsymbol{\delta}_B + \boldsymbol{N}\boldsymbol{x}_N^i + \boldsymbol{N}\boldsymbol{\delta}_N = \boldsymbol{b}$$

$$\Rightarrow \quad \boldsymbol{B}\boldsymbol{x}_B^i + \boldsymbol{N}\boldsymbol{x}_N^i + \boldsymbol{B}\boldsymbol{\delta}_B + \boldsymbol{N}\boldsymbol{\delta}_N = \boldsymbol{b}$$

$$\Rightarrow \quad \boldsymbol{b} + \boldsymbol{B}\boldsymbol{\delta}_B + \boldsymbol{N}\boldsymbol{\delta}_N = \boldsymbol{b} \qquad \Rightarrow \qquad \boxed{\boldsymbol{\delta}_B = -\boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{\delta}_N}$$

That is, if we change $\boldsymbol{x}_B$ by $\boldsymbol{\delta}_B = -\boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{\delta}_N$ then our new design will satisfy $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$.

Now consider Δf (the change in f) for small $\boldsymbol{\delta}$: $\Delta f \cong \nabla \boldsymbol{f}^T(\boldsymbol{x}^i)\cdot\boldsymbol{\delta}$,

$$\Rightarrow \quad \Delta f = \nabla_B \boldsymbol{f}^T(\boldsymbol{x}^i)\cdot\boldsymbol{\delta}_B + \nabla_N \boldsymbol{f}^T(\boldsymbol{x}^i)\cdot\boldsymbol{\delta}_N \qquad (\otimes)$$

Substituting for $\boldsymbol{\delta}_B$ in ($\otimes$) above we have:

$$\Delta f = \nabla_B \boldsymbol{f}^T(\boldsymbol{x}^i)\cdot(-\boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{\delta}_N) + \nabla_N \boldsymbol{f}^T(\boldsymbol{x}^i)\cdot\boldsymbol{\delta}_N = \{\nabla_N \boldsymbol{f}^T(\boldsymbol{x}^i) - \nabla_B \boldsymbol{f}^T(\boldsymbol{x}^i)\boldsymbol{B}^{-1}\boldsymbol{N}\}\boldsymbol{\delta}_N = [\boldsymbol{\gamma}_N]^T\boldsymbol{\delta}_N$$

(say), where $[\boldsymbol{\gamma}_N]^T = \nabla_N \boldsymbol{f}^T(\boldsymbol{x}^i) - \nabla_B \boldsymbol{f}^T(\boldsymbol{x}^i)\boldsymbol{B}^{-1}\boldsymbol{N}$

In summary:

IF $\boldsymbol{x}_N$ changes from $\boldsymbol{x}_N^{\,i}$ by $\boldsymbol{\delta}_N$,

AND $\boldsymbol{x}_B$ is correspondingly changed from $\boldsymbol{x}_B^{\,i}$ by $\boldsymbol{\delta}_B$ to maintain feasibility,

THEN Δf is the change in f.

Now, since we are minimizing, we would like to move in the direction $-\gamma_N$ to decrease f. Let $\boldsymbol{\Gamma}_N$ be the search direction for $\boldsymbol{x}_N$. Then for $i \in N$,

a) If $\gamma_i \le 0$, $\Gamma_i = -\gamma_i\ (\ge 0)$: we 'increase' x_i in the direction $-\Gamma_i$

b) If $\gamma_i \ge 0$, $\Gamma_i = -\gamma_i\ (\le 0)$: we 'decrease' x_i in the direction $-\Gamma_i$ **<u>as long as</u>** $x_i > 0$,

$= 0$ if $x_i = 0$ (since cannot decrease below zero…)

As proposed initially by Wolfe, the *Reduced Gradient* method uses for $i \in N$

$$\Gamma_i = \begin{cases} 0 \text{ if } \gamma_i \ge 0 \ \&\ x_i = 0 \\ -\gamma_i \text{ otherwise} \end{cases}$$

The *Convex Simplex* method (Zangwill) chooses only <u>one</u> axis to move along. Let,

$\alpha = \text{Max}\ \{-\gamma_i \mid \gamma_i \le 0\}$ case (a),

$\beta = \text{Max}\ \{x_i\gamma_i \mid \gamma_i \ge 0\}$ case (b),

$\Rightarrow$ $\alpha = -\gamma_s\ (\ge 0)$, $\beta = x_t\gamma_t\ (\ge 0)$

If

(1) $\alpha \geq \beta$ OR $\alpha > 0$, $\beta = \phi$ then,

$\boldsymbol{\Gamma}_N = [0 \;\; 0 \ldots 0 \;\; 1 \;\; 0 \ldots 0]^T$ s^{th} position

(2) $\beta > \alpha$ OR $\beta > 0$, $\alpha = \phi$ then,

$\boldsymbol{\Gamma}_N = [0 \;\; 0 \ldots 0 \;\; -1 \;\; 0 \ldots 0]^T$ t^{th} position

(3) $\beta = \alpha = 0$ OR $\alpha = 0$, $\beta = \phi$ OR $\alpha = \phi$, $\beta = \phi$ then,

we have a **K-K-T** point: **STOP!**

Once we have $\boldsymbol{\Gamma}_N$, then $\boldsymbol{\Gamma}_B = -\boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{\Gamma}_N$ and $\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{\Gamma}_B \\ \hdashline \boldsymbol{\Gamma}_N \end{bmatrix}$

Once $\boldsymbol{\Gamma}$ is determined we do a line search as usual in the direction $\boldsymbol{\Gamma}$ (by using a one dimensional search procedure like Golden Section , Fibonacci etc.), i.e.,,

Min $f(\boldsymbol{x}^i + \alpha\boldsymbol{\Gamma})$, st $0 \leq \alpha \leq \alpha_{max}$, where $\alpha_{max} = \sup \{\alpha \mid \boldsymbol{x}^i + \alpha\boldsymbol{\Gamma} \geq 0\}$

If this yields α^*, then $\boldsymbol{x}^{i+1} = \boldsymbol{x}^i + \alpha^*\boldsymbol{\Gamma}$ and so on...

EXAMPLE (*CSM*):

Min $x_1^2 - x_1 - x_2$

st $2x_1 + x_2 \leq 1$,

$x_1 + 2x_2 \leq 1$, $x_1, x_2 \geq 0$

Adding slack variables we rewrite the constraints as

$2x_1 + x_2 + x_3 = 1$

$x_1 + 2x_2 + x_4 = 1$; $x_1, x_2, x_3, x_4 \geq 0$

For this problem, $\nabla \boldsymbol{f}^{\mathrm{T}}(\boldsymbol{x}) = \begin{bmatrix} 2x_1 - 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}$.

Start with $x_1 = x_2 = 0$, $x_3 = x_4 = 1$, $\boldsymbol{f}(\boldsymbol{x^0})=0$, and with x_3 and x_4 in the basis.

ITERATION 1: $\boldsymbol{x}_{\mathrm{B}} = \begin{bmatrix} x_3 \\ x_4 \end{bmatrix}, \quad \boldsymbol{x}_{\mathrm{N}} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_{\mathrm{B}} \\ \boldsymbol{x}_{\mathrm{N}} \end{bmatrix} = \begin{bmatrix} x_3 \\ x_4 \\ x_1 \\ x_2 \end{bmatrix}$

$\boldsymbol{A} = [\boldsymbol{B} \;\vdots\; \boldsymbol{N}] = \left[\begin{array}{cc:cc} 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 2 \end{array}\right]$. (Note that Rank($\boldsymbol{B}$)=2)

$\boldsymbol{x}_{\mathrm{B}}^{\boldsymbol{0}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{x}_{\mathrm{N}}^{\boldsymbol{0}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, and $\boldsymbol{x^0} = [0 \quad 0 \quad 1 \quad 1]^{\mathrm{T}}$

$$\nabla_B \boldsymbol{f}(\boldsymbol{x^0}) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \text{and } \nabla_{\mathrm{N}} \boldsymbol{f}(\boldsymbol{x^0}) = \begin{bmatrix} \partial f/\partial x_1 \\ \partial f/\partial x_2 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

Search Direction: $\gamma_{\mathrm{N}}{}^{\mathrm{T}} = \nabla_{\mathrm{N}} \boldsymbol{f}^{\mathrm{T}}(\boldsymbol{x^0}) - \nabla_{\mathrm{B}} \boldsymbol{f}^{\mathrm{T}}(\boldsymbol{x^0}) \boldsymbol{B}^{-1} \boldsymbol{N}$

$$\Rightarrow \quad \boldsymbol{\gamma}_{\mathrm{N}}^{\mathrm{T}} = [-1 \;\; -1] - [0 \quad 0] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{\gamma}_{\mathrm{N}} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

$\alpha = max\{+1, +1\} = 1$; $\beta = \phi$

We could choose $\Gamma_{\mathrm{N}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ or $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$. Let us choose $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$...

$$\boldsymbol{\Gamma}_{\mathrm{B}} = -\boldsymbol{B}^{-1} \boldsymbol{N} \boldsymbol{\Gamma}_{\mathrm{N}} = -\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -2 \\ -1 \end{bmatrix}; \quad \therefore \boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{\Gamma}_{\mathrm{B}} \\ \boldsymbol{\Gamma}_{\mathrm{N}} \end{bmatrix} = \begin{bmatrix} -2 \\ -1 \\ 1 \\ 0 \end{bmatrix} \approx \begin{matrix} x_3 \\ x_4 \\ x_1 \\ x_2 \end{matrix}$$

Line Search: $\text{Minimize}_{0 \le \alpha \le \alpha_{max}} f\left\{ \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} + \alpha \begin{bmatrix} 1 \\ 0 \\ -2 \\ -1 \end{bmatrix} \right\} = f\left\{ \begin{bmatrix} \alpha \\ 0 \\ 1 - 2\alpha \\ 1 - \alpha \end{bmatrix} \right\}$

$\alpha_{max} = \sup\{ \alpha \mid \alpha \ge 0, \; - \; , \; (1-2\alpha) \ge 0, (1-\alpha) \ge 0 \} = 0.5$

$\Rightarrow \text{Minimize}_{0\leq\alpha\leq 0.5}\ \{\alpha^2 - \alpha - 0\}$

A line search yields α^*=0.5. Therefore

$$\boldsymbol{x^1} = \boldsymbol{x^0} + \alpha^*\boldsymbol{\Gamma} = \begin{bmatrix} 1/2 \\ 0 \\ 0 \\ 1/2 \end{bmatrix},\ f(\boldsymbol{x^1}) = -0.25$$

In summary, x_1 has now entered the basis and replaced x_3 (which cannot be in the basis since it has reached its lower bound of zero).

ITERATION 2: $\boldsymbol{x}_\text{B} = \begin{bmatrix} x_1 \\ x_4 \end{bmatrix}, \quad \boldsymbol{x}_\text{N} = \begin{bmatrix} x_3 \\ x_2 \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_\text{B} \\ \boldsymbol{x}_\text{N} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_4 \\ x_3 \\ x_2 \end{bmatrix}$

$$\boldsymbol{A} = [\boldsymbol{B} \mid \boldsymbol{N}] = \left[\begin{array}{cc|cc} 2 & 0 & 1 & 1 \\ 1 & 1 & 0 & 2 \end{array}\right], \quad \text{and } B^{-1} = \begin{bmatrix} 0.5 & 0 \\ -0.5 & 1 \end{bmatrix}$$

$$\boldsymbol{x}_\text{B}^\mathbf{1} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \quad \boldsymbol{x}_\text{N}^\mathbf{1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \text{and } \boldsymbol{x^1} = [0.5 \quad 0.5 \quad 0 \quad 0]^\text{T}$$

$$\boldsymbol{\nabla}_B \boldsymbol{f}(\boldsymbol{x^1}) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \text{and } \boldsymbol{\nabla}_\text{N}\boldsymbol{f}(\boldsymbol{x^1}) = \begin{bmatrix} \partial f/\partial x_1 \\ \partial f/\partial x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

Search Direction: $\boldsymbol{\gamma}_\text{N}^\text{T} = \boldsymbol{\nabla}_\text{N}\boldsymbol{f}^\text{T}(\boldsymbol{x^1}) - \boldsymbol{\nabla}_\text{B}\boldsymbol{f}^\text{T}(\boldsymbol{x^1})\boldsymbol{B}^{-1}\boldsymbol{N}$

$$\Rightarrow \quad \boldsymbol{\gamma}_\text{N}^\text{T} = [0 \quad -1] - [0 \quad 0] \begin{bmatrix} 0.5 & 0 \\ -0.5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{\gamma}_\text{N} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

$\alpha = max\{+1\} = 1;$ $\beta = max\ (0 \cdot 0) = 0$

Since $\alpha > \beta$ we have $\boldsymbol{\Gamma}_\text{N} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

$$\boldsymbol{\Gamma}_\text{B} = -\boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{\Gamma}_\text{N} = -\begin{bmatrix} 0.5 & 0 \\ -0.5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.5 \\ -1.5 \end{bmatrix}; \quad \therefore \boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{\Gamma}_\text{B} \\ \boldsymbol{\Gamma}_\text{N} \end{bmatrix} = \begin{bmatrix} -0.5 \\ -1.5 \\ 0 \\ 1 \end{bmatrix} \approx \begin{matrix} x_1 \\ x_4 \\ x_3 \\ x_2 \end{matrix}$$

Line Search: $\text{Minimize}_{0\le\alpha\le\alpha_{max}} f\left\{\begin{bmatrix}0.5\\0\\0\\0.5\end{bmatrix}+\alpha\begin{bmatrix}-0.5\\1\\0\\-1.5\end{bmatrix}\right\}=f\left\{\begin{bmatrix}(1-\alpha)/2\\\alpha\\0\\(1-3\alpha)/2\end{bmatrix}\right\}$

α_{max}=sup{ α | (1-α)/2≥0, α≥0, - , (1-3α)/2≥0 } = 1/3

$\Rightarrow \text{Minimize}_{0\le\alpha\le 1/3}\ \{[(1-\alpha)/2]^2-[(1-\alpha)/2]-\alpha\}$

A line search yields α^*=1/3. Therefore

$\boldsymbol{x^2}=\boldsymbol{x^1}+\alpha^*\boldsymbol{\Gamma}=\begin{bmatrix}1/3\\1/3\\0\\0\end{bmatrix}$, $f(\boldsymbol{x}^2)$= -0.555

Thus x_2 has now entered the basis and replaced x_4

ITERATION 3: $\boldsymbol{x}_\text{B}=\begin{bmatrix}x_1\\x_2\end{bmatrix}$, $\boldsymbol{x}_\text{N}=\begin{bmatrix}x_3\\x_4\end{bmatrix}$, $\boldsymbol{x}=\begin{bmatrix}\boldsymbol{x}_\text{B}\\\hdashline\boldsymbol{x}_\text{N}\end{bmatrix}=\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}$

$\boldsymbol{A}=[\boldsymbol{B}\ \vdots\ \boldsymbol{N}]=\left[\begin{array}{cc:cc}2&1&1&0\\1&2&0&1\end{array}\right]$, and $B^{-1}=\begin{bmatrix}2/3&-1/3\\-1/3&2/3\end{bmatrix}$

$\boldsymbol{x}_\text{B}^{\boldsymbol{2}}=\begin{bmatrix}1/3\\1/3\end{bmatrix}$, $\boldsymbol{x}_\text{N}^{\boldsymbol{2}}=\begin{bmatrix}0\\0\end{bmatrix}$, and $\boldsymbol{x^2}=[1/3\quad 1/3\quad 0\quad 0]^\text{T}$

$\boldsymbol{\nabla}_B\boldsymbol{f}(\boldsymbol{x^2})=\begin{bmatrix}-1/3\\-1\end{bmatrix}$, and $\boldsymbol{\nabla}_\text{N}\boldsymbol{f}(\boldsymbol{x^2})=\begin{bmatrix}\partial f/\partial x_3\\\partial f/\partial x_4\end{bmatrix}=\begin{bmatrix}0\\0\end{bmatrix}$

Search Direction: $\boldsymbol{\gamma}_\text{N}^\text{T}=\boldsymbol{\nabla}_\text{N}\boldsymbol{f}^\text{T}(\boldsymbol{x^0})-\boldsymbol{\nabla}_\text{B}\boldsymbol{f}^\text{T}(\boldsymbol{x^0})\boldsymbol{B}^{-1}\boldsymbol{N}$

$\Rightarrow\ \boldsymbol{\gamma}_\text{N}^\text{T}=[0\quad 0]-[-1/3\quad -1]\begin{bmatrix}2/3&-1/3\\-1/3&2/3\end{bmatrix}\begin{bmatrix}1&0\\0&1\end{bmatrix}\ \Rightarrow\quad \boldsymbol{\gamma}_\text{N}=\begin{bmatrix}-1/9\\5/9\end{bmatrix}$

$\alpha = max\{+1/9\} = 1/9$; $\beta=max\ (0\cdot 5/9)=0$

Since $\alpha > \beta$ we have $\boldsymbol{\Gamma}_N = \begin{bmatrix}1\\0\end{bmatrix}$.

$$\boldsymbol{\Gamma}_B = -\boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{\Gamma}_N = -\begin{bmatrix}2/3 & -1/3\\ -1/3 & 2/3\end{bmatrix}\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}-2/3\\1/3\end{bmatrix};$$

$$\therefore \boldsymbol{\Gamma} = \begin{bmatrix}\boldsymbol{\Gamma}_B\\ \boldsymbol{\Gamma}_N\end{bmatrix} = \begin{bmatrix}-2/3\\1/3\\1\\0\end{bmatrix} \approx \begin{matrix}x_1\\x_2\\x_3\\x_4\end{matrix}$$

Line Search: $\text{Minimize}_{0\le\alpha\le\alpha_{max}} f\left\{\begin{bmatrix}1/3\\1/3\\0\\0\end{bmatrix} + \alpha\begin{bmatrix}-2/3\\1/3\\1\\0\end{bmatrix}\right\} = f\left\{\begin{bmatrix}(1-2\alpha)/3\\(1+\alpha)/3\\\alpha\\0\end{bmatrix}\right\}$

α_{max}=sup{ α | (1-2α)≥0, 1+α)≥0, α≥0, - } = 0.5

$\Rightarrow \text{Minimize}_{0\le\alpha\le0.5}\ \{[(1-2\alpha)/3]^2 - [(1-2\alpha)/3] - (1+\alpha)/3\}$

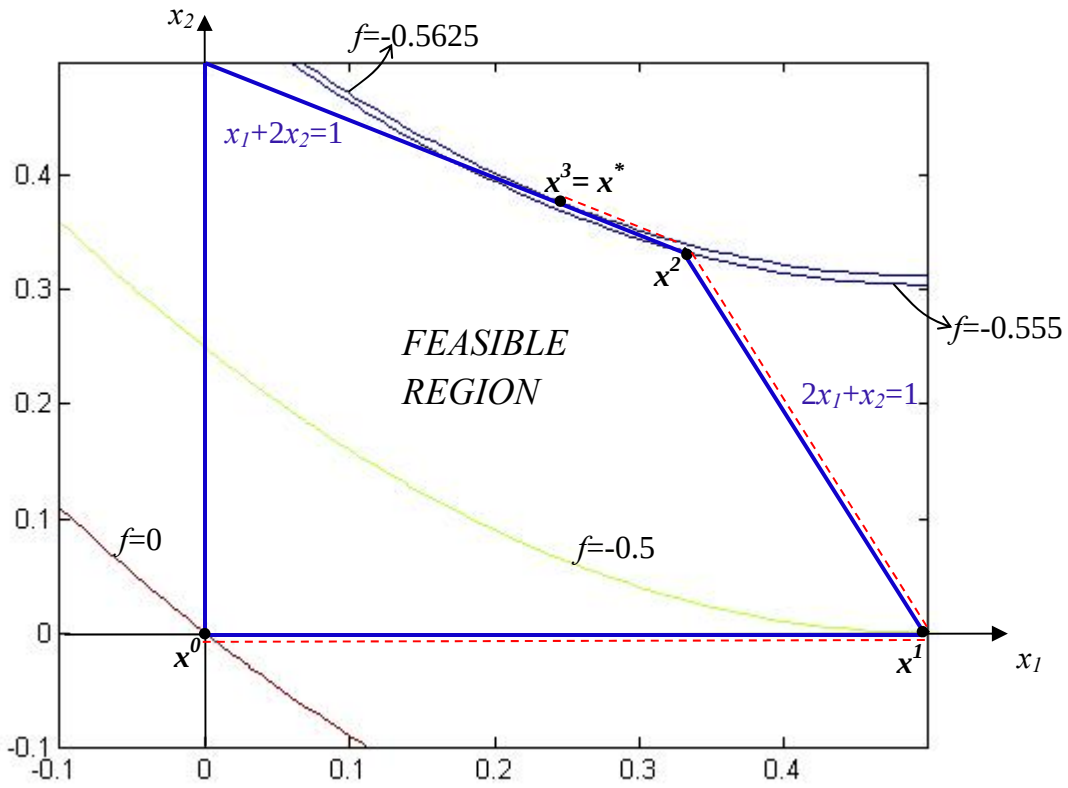
A line search yields α^*=1/8. Therefore

$$\boldsymbol{x}^3 = \boldsymbol{x}^2 + \alpha^*\boldsymbol{\Gamma} = \begin{bmatrix}1/4\\3/8\\1/8\\0\end{bmatrix},\ f(\boldsymbol{x}^3) = -0.5625$$

At this point x_4 is definitely nonbasic, we could select any two of the other three to be basic as long as they have linearly independent columns in $\boldsymbol{A}$; let us (arbitrarily) choose x_1 and x_2 (same as before).

ITERATION 4: Continue and verify that $\boldsymbol{x}^3$ is the optimum solution...

The algorithm's progress is as shown below:



Path followed by the algorithm

GENERALIZED REDUCED GRADIENT (GRG) ALGORITHM

GRG (Abadie and Carpentier) is essentially a generalization of the convex simplex method to solve

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

$$\text{st} \quad g_j(\boldsymbol{x}) = 0, \quad j=1,2,...,m,$$

$$a_i \le x_i \le b_i, \quad i=1,2,...,n; \quad \boldsymbol{x} \in \mathbb{R}^n.$$

GRG is considered to be one of the better algorithms for solving problems of the above form.

SIMILAR IN CONCEPT TO SIMPLEX METHOD FOR LP....

SIMPLEX		GRG
Reduced Costs	≈	Reduced Gradients
Basic Variables	≈	Dependent Variables
Nonbasic Variables	≈	Independent Variables

DIFFERS IN THAT

- Nonbasic (independent) variables need not be at their bounds,
- At each iteration, several nonbasic (independent) variables could change values
- Basis need not change from one iteration to the next.

At the beginning of each iteration we partition the *n* decision variables into 2 sets:

1) **D**ependent variables (one per equation),

2) **I**ndependent variables

Let $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_{\mathrm{D}} \\ \hdashline \boldsymbol{x}_{\mathrm{I}} \end{bmatrix}$ (after reordering the variables)

Here, $\boldsymbol{x}_{\mathrm{D}}$ is a vector of (*m*) **D**ependent variables

$\boldsymbol{x}_{\mathrm{I}}$ is a vector of (*n-m*) **I**ndependent variables

Similarly, we partition

$$\boldsymbol{a} = \begin{bmatrix} \boldsymbol{a}_{\mathrm{D}} \\ \boldsymbol{a}_{\mathrm{I}} \end{bmatrix}; \quad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_{\mathrm{D}} \\ \boldsymbol{b}_{\mathrm{I}} \end{bmatrix}; \quad \nabla \boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \nabla_{\mathrm{D}} \boldsymbol{f}(\boldsymbol{x}) \\ \nabla_{\mathrm{I}} \boldsymbol{f}(\boldsymbol{x}) \end{bmatrix} \begin{matrix} m \text{ rows} \\ (n-m) \text{ rows} \end{matrix}$$

The Jacobian matrix (***J***) is

$$\boldsymbol{J}(\boldsymbol{x}) = \left[\begin{array}{c|c} \nabla_{\mathrm{D}}^{\mathrm{T}} \boldsymbol{g_1}(\boldsymbol{x}) & \nabla_{\mathrm{I}}^{\mathrm{T}} \boldsymbol{g_1}(\boldsymbol{x}) \\ \nabla_{\mathrm{D}}^{\mathrm{T}} \boldsymbol{g_2}(\boldsymbol{x}) & \nabla_{\mathrm{I}}^{\mathrm{T}} \boldsymbol{g_2}(\boldsymbol{x}) \\ \vdots & \vdots \\ \vdots & \vdots \\ \nabla_{\mathrm{D}}^{\mathrm{T}} \boldsymbol{g_m}(\boldsymbol{x}) & \nabla_{\mathrm{I}}^{\mathrm{T}} \boldsymbol{g_m}(\boldsymbol{x}) \end{array}\right] = [\boldsymbol{J}_{\mathrm{D}} \mid \boldsymbol{J}_{\mathrm{I}}]$$, where $\boldsymbol{J}_{\mathrm{D}}$ is *m*×*m* and $\boldsymbol{J}_{\mathrm{I}}$ is *m*×(*n-m*)

m cols. (*n-m*) cols.

Note that ***J*** has *m* rows; row *i* is $\nabla \boldsymbol{g_i}^{\mathrm{T}}(\boldsymbol{x})$.

Let $\boldsymbol{x^i} = \begin{bmatrix} \boldsymbol{x}_{\mathrm{D}}^{\boldsymbol{i}} \\ \hdashline \boldsymbol{x}_{\mathrm{I}}^{\boldsymbol{i}} \end{bmatrix}$ be our current design, where

- $g_j(\boldsymbol{x^i}) = 0 \ \forall \ j$, ($\boldsymbol{x^i}$ is feasible)
- $\boldsymbol{a}_{\mathrm{D}} < \boldsymbol{x}_{\mathrm{D}}^{\boldsymbol{i}} < \boldsymbol{b}_{\mathrm{D}}$ (dependent variables are not <u>at</u> a bound)
- $\boldsymbol{J}_{\mathrm{D}}$ is nonsingular ($\boldsymbol{J}_{\mathrm{D}}^{-1}$ exists)
- $\boldsymbol{a}_{\mathrm{I}} \leq \boldsymbol{x}_{\mathrm{I}}^{\boldsymbol{i}} \leq \boldsymbol{b}_{\mathrm{I}}$ (independent variables are <u>within</u> their bounds)

Let $\boldsymbol{\delta} = \begin{bmatrix} \boldsymbol{\delta}_D \\ \hdashline \boldsymbol{\delta}_I \end{bmatrix}$ be a change vector in the value of $\boldsymbol{x}^i$

For small values of $\boldsymbol{\delta}$, Δg_j = change in $g_j \cong \nabla \boldsymbol{g_j}^T(\boldsymbol{x}^i)\cdot\boldsymbol{\delta}$.

Choose $\boldsymbol{\delta}$ so that $\boldsymbol{x}^{i+1}$ is still feasible (or to be exact, near-feasible), i.e., choose $\boldsymbol{\delta}$ э

$$\Delta g_j \cong \nabla \boldsymbol{g}_j^T(\boldsymbol{x}^i)\cdot\boldsymbol{\delta} = 0, \qquad j = 1,2,\ldots,m$$

$$\Rightarrow \Delta g_j \cong \nabla_D \boldsymbol{g}_j^T(\boldsymbol{x}^i)\cdot\boldsymbol{\delta}_D + \nabla_I \boldsymbol{g}_j^T(\boldsymbol{x}^i)\cdot\boldsymbol{\delta}_I = 0 \quad \forall\, j$$

$$\Rightarrow \Delta \boldsymbol{g} = \begin{bmatrix} \Delta g_1 \\ \Delta g_2 \\ \vdots \\ \vdots \\ \Delta g_m \end{bmatrix} = \boldsymbol{J}(\boldsymbol{x}^i)\boldsymbol{\delta} = [\boldsymbol{J}_D(\boldsymbol{x}^i)]\boldsymbol{\delta}_D + [\boldsymbol{J}_I(\boldsymbol{x}^i)]\boldsymbol{\delta}_I = \boldsymbol{0}$$

Since $\boldsymbol{J}_D(\boldsymbol{x}^i)$ is nonsingular, $[\boldsymbol{J}_D(\boldsymbol{x}^i)]^{-1}$ exists and

$$\boxed{\boldsymbol{\delta}_D = -\,[\boldsymbol{J}_D(\boldsymbol{x}^i)]^{-1}[\boldsymbol{J}_I(\boldsymbol{x}^i)]\boldsymbol{\delta}_I}$$

Similarly, for small values of $\boldsymbol{\delta}$, Δf = change in $f \cong \nabla \boldsymbol{f}^T(\boldsymbol{x}^i)\cdot\boldsymbol{\delta}$

$$\Rightarrow \quad \Delta f = \nabla_D \boldsymbol{f}^T(\boldsymbol{x}^i)\cdot\boldsymbol{\delta}_D + \nabla_I \boldsymbol{f}^T(\boldsymbol{x}^i)\cdot\boldsymbol{\delta}_I \qquad (\otimes)$$

Substituting $\boldsymbol{\delta}_D$ in ($\otimes$) we have

$$\Delta f = \nabla_D \boldsymbol{f}^T(\boldsymbol{x}^i)\,\{-[\boldsymbol{J}_D(\boldsymbol{x}^i)]^{-1}[\boldsymbol{J}_I(\boldsymbol{x}^i)]\boldsymbol{\delta}_I\} + \nabla_I \boldsymbol{f}^T(\boldsymbol{x}^i)\cdot\boldsymbol{\delta}_I$$

$$= \{\, \nabla_I \boldsymbol{f}^T(\boldsymbol{x}^i) - \nabla_D \boldsymbol{f}^T(\boldsymbol{x}^i)\,[\boldsymbol{J}_D(\boldsymbol{x}^i)]^{-1}[\boldsymbol{J}_I(\boldsymbol{x}^i)]\,\}\cdot\boldsymbol{\delta}_I$$

i.e., $\boxed{\Delta f = \boldsymbol{\gamma}_i^T \boldsymbol{\delta}_I}$

where

$$\boxed{\gamma_I = \nabla_I \boldsymbol{f}(\boldsymbol{x}^i) - \left[\nabla_D \boldsymbol{f}^T(\boldsymbol{x}^i)\,[\boldsymbol{J}_D(\boldsymbol{x}^i)]^{-1}[\boldsymbol{J}_I(\boldsymbol{x}^i)]\right]^T}$$

is called the **REDUCED GRADIENT VECTOR**.

In other words…

IF $\boldsymbol{x}_{\mathrm{I}}^{\boldsymbol{i}}$ is changed by an amount $\boldsymbol{\delta}_{\mathrm{I}}$, AND $\boldsymbol{x}_{\mathrm{D}}^{\boldsymbol{i}}$ is changed by an amount $\boldsymbol{\delta}_{\mathrm{D}}$ in order to maintain $g_j(\boldsymbol{x}) \cong 0$ for all j, THEN $\Delta f = \boldsymbol{\gamma}_{\boldsymbol{i}}^{\mathrm{T}} \boldsymbol{\delta}_{\mathrm{I}}$ is the corresponding change in the value of f.

Since we are minimizing we want Δf to be negative. Assuming that $\delta_i > 0$, (so that $-\delta_i$ would correspond to a decrease in x_i and $+\delta_i$ to an increase), we would therefore move the negative of the direction corresponding to the vector $\boldsymbol{\gamma}_{\mathrm{I}}$ while ensuring that the bounds on x_i (namely a_i and b_i) are not violated. Thus the STEP DIRECTION $\boldsymbol{\Gamma}_{\mathrm{I}}$ for the independent variables has elements given by

$$(\boldsymbol{\Gamma}_{\mathrm{I}})_i = \begin{cases} -\gamma_i \text{ if } a_i < x_i < b_i \quad (can\ go\ in\ either\ direction) \\ -\gamma_i \text{ if } \gamma_i > 0 \text{ and } x_i = b_i \\ -\gamma_i \text{ if } \gamma_i < 0 \text{ and } x_i = a_i \\ 0 \quad \text{if } \gamma_i > 0 \text{ and } x_i = a_i \quad (can't\ decrease\ x_i \ldots) \\ 0 \quad \text{if } \gamma_i < 0 \text{ and } x_i = b_i \quad (can't\ increase\ x_i \ldots) \end{cases}$$

For the dependent variables,

$$\boldsymbol{\Gamma}_{\mathrm{D}} = -[\boldsymbol{J}_{\mathrm{D}}(\boldsymbol{x}^{\boldsymbol{i}})]^{-1} [\boldsymbol{J}_{\mathrm{I}}(\boldsymbol{x}^{\boldsymbol{i}})]\, \boldsymbol{\Gamma}_{\mathrm{I}}$$

can go either way irrespective of the direction of $\boldsymbol{\Gamma}_{\mathrm{I}}$ since $\boldsymbol{a}_{\mathrm{D}} < \boldsymbol{x}_{\mathrm{D}} < \boldsymbol{b}_{\mathrm{D}}$.

Contrast this with the Simplex method for LP where only one independent (nonbasic) variable is changed at each iteration by selecting Max $|\gamma_i|$ which doesn't violate a constraint (≈ the ratio test for the minimum value…).

Having found the search direction $\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{\Gamma}_D \\ \hdashline \boldsymbol{\Gamma}_I \end{bmatrix}$ we do the usual line search (using one of Golden Section, Fibonacci, etc.) to find the step size, i.e.,

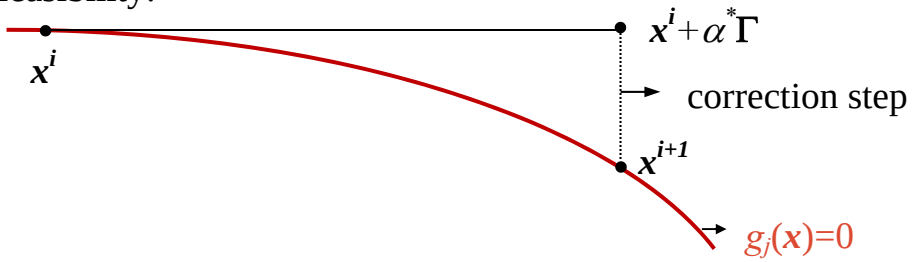
$$\text{Min } f(\boldsymbol{x}^i + \alpha\boldsymbol{\Gamma}), \text{ st } \quad 0 \le \alpha \le \alpha_{max}, \quad \text{where } \alpha_{max} = \sup\{\alpha \mid \boldsymbol{a} \le \boldsymbol{x}^i + \alpha\boldsymbol{\Gamma} \le \boldsymbol{b}\}$$

Suppose the optimum step size is α^*.

Correction Step

In general, we might find that $g_j(\boldsymbol{x}^i + \alpha^*\boldsymbol{\Gamma}) \neq 0$ for some j.

Therefore, we need a correction step (as in the gradient projection method) to re-attain feasibility:



This is done for example, by fixing the independent variables at the values found and then solving the system $\boldsymbol{g}(\boldsymbol{x})=\boldsymbol{0}$ using $\boldsymbol{x}^i + \alpha^*\boldsymbol{\Gamma}$ as the initial guess for a Newton-Raphson search.

Suppose we write $\boldsymbol{x}^i + \alpha^*\boldsymbol{\Gamma} = [\boldsymbol{x}_I\ ,\ \boldsymbol{x}_D]^T$. Then our problem at this stage is to solve $\boldsymbol{g}(\boldsymbol{x}_I, \boldsymbol{y})=\boldsymbol{0}$ by adjusting only $\boldsymbol{y}$ (a system of m equations in m unknowns).

We start with $\boldsymbol{y}^0 = \boldsymbol{x}_D$, and Newton's iterations are applied for r=1,2,… via

$$\boldsymbol{y}^{r+1} = \boldsymbol{y}^r - [\boldsymbol{J}_D(\boldsymbol{x}_I, \boldsymbol{y}^r)]^{-1}\boldsymbol{g}(\boldsymbol{x}_I, \boldsymbol{y}^r)$$

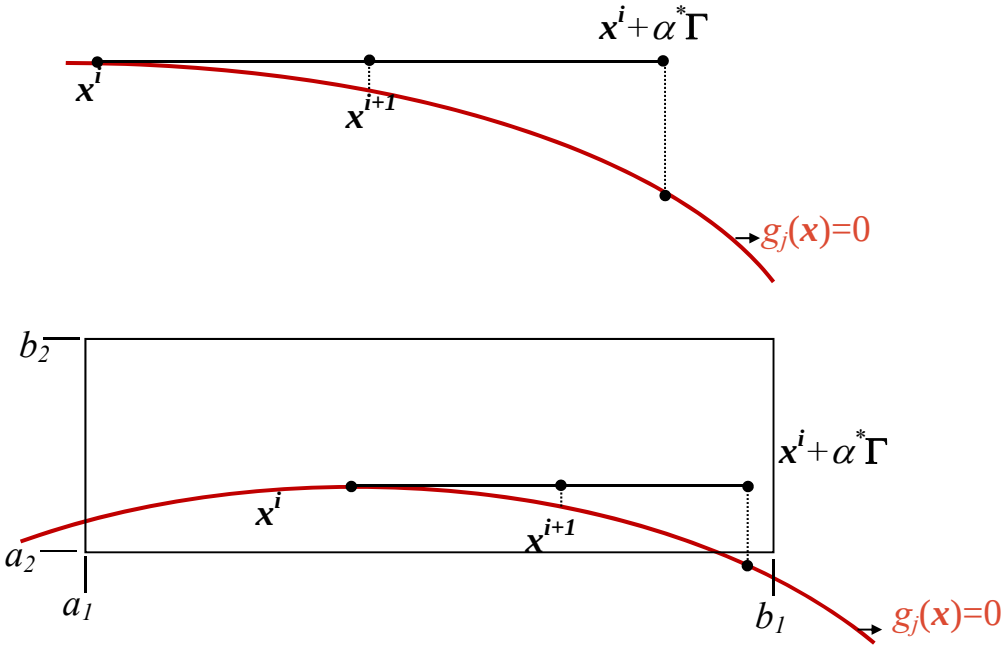
until $\boldsymbol{g}(\boldsymbol{x}_I, \boldsymbol{y}^k)=\boldsymbol{0}$ for some k. We then set $\boldsymbol{x}_D = \boldsymbol{y}^k$.

Here, $\boldsymbol{J}_D(\boldsymbol{x}_I, \boldsymbol{y}^r)$ is the Jacobian matrix for the constraint functions evaluated at $\boldsymbol{y}=\boldsymbol{y}^r$ for the dependent variables, with the independent variables fixed at $\boldsymbol{x}_I$

There are three possible pitfalls during the correction step that we must safeguard against:

1. The final feasible solution $\boldsymbol{x}^{i+1}=(\boldsymbol{x}_I, \boldsymbol{x}_D)$ might yield a value for f that is worse than what it was at $\boldsymbol{x}^i$.
2. One or more bounds on $\boldsymbol{x}$ are violated during the Newton's iterations
3. The constraint violation actually starts to *increase*.

In all three cases, the typical solution is to reduce the step size along the direction $\boldsymbol{\Gamma}$ from α^* to (say) $\alpha^*/2$. Now the point $\boldsymbol{x}^i+\alpha^*\boldsymbol{\Gamma}$ is closer to $\boldsymbol{x}^i$ and the Newton iterations are re-executed to obtain a new feasible point $\boldsymbol{x}^{i+1}$ with $f(\boldsymbol{x}^{i+1})< f(\boldsymbol{x}^i)$. This process might need to be repeated more than once…



Finding Search Directions from the Reduced Gradient vector γ

There are several versions actually for picking Γ_I. Let us start by defining

$$S = \left\{ j \mid (\gamma_I)_j > 0 \ \& \ (x_I)_j = (a_I)_j, \ \text{OR} \ \ (\gamma_I)_j < 0 \ \& \ (x_I)_j = (b_I)_j \right\}$$

Note that S is an index subset of I (the index set of independent variables), denoting variables that (1) we would like to decrease but are already at their lower bounds, or (2) we would like to increase but are already at their upper bounds.

Version 1: $(\Gamma_I)_j = \begin{cases} 0 \text{ if } \ j \in S \\ -(\gamma_I)_j \text{ otherwise} \end{cases}$ (this is the classical version)

Version 2: Identify $s \in$ *argmax* $\{\text{Max}_{j \notin S} |(\gamma_I)_j|\}$ and set

$$(\Gamma_I)_j = \begin{cases} 0 \text{ if } \ j \neq s \\ -(\gamma_I)_s \ \textit{if} \ \ j = s \end{cases} \qquad \text{(GRGS)}$$

This version coincides with the "greedy" version of the simplex method for LP if the optimization problem is a linear program.

Version 3: Cyclically set $(\Gamma_I)_j = -(\gamma_I)_j$ for $j \notin S$ and all other $(\Gamma_I)_j = 0$, where a cycle consists of n iterations. (GRGC)

In other words, for k=1,2,…,n set

$$(\Gamma_I)_j^k = \begin{cases} 0 \text{ if } \ j \neq k \\ -(\gamma_I)_j \ \textit{if} \ \ j = k \end{cases}$$

as long as $k \notin S$ and k is not a basic variable. If k is basic or is in S we simply move on to k+1 instead. After k=n we return to k=1 and continue the process.

GRG and the K-K-T Conditions

Recall that with GRG we stop when (*i*) $\gamma_i \le 0$ for all $\{i \ni x_i=b_i\}$, (*ii*) $\gamma_i \ge 0$, for all $\{i \ni x_i=a_i\}$, and (*iii*)$\gamma_I = 0$ for all $\{i \ni x_i<b_i$ and $x_i>a_i\}$.

This stopping condition is related to the K-K-T conditions for the original problem:

Min $f(\boldsymbol{x})$

st $g_j(\boldsymbol{x})=0$ for $j=1,2,\ldots,m$

$a_i \le x_i \le b_i$ for $i=1,2,\ldots,n$

Consider the Lagrangian for the above problem:

$$L(\boldsymbol{x},\boldsymbol{\lambda}) = f(\boldsymbol{x}) + \sum_j \lambda_j g_j(\boldsymbol{x}) - \sum_i \alpha_i(x_i-a_i) + \sum_i \beta_i(x_i-b_i)$$

At the optimum the K-K-T conditions yield

$\nabla \boldsymbol{L} = \nabla \boldsymbol{f}(\boldsymbol{x}) + \sum_j \lambda_j \nabla \boldsymbol{g_j}(\boldsymbol{x}) - (\boldsymbol{\alpha} - \boldsymbol{\beta}) = \boldsymbol{0}$, along with the original constraints; $\boldsymbol{\alpha},\boldsymbol{\beta} \ge \boldsymbol{0}$, and complementary slackness for the bounding constraints. Thus

$\nabla \boldsymbol{L} = \nabla \boldsymbol{f}(\boldsymbol{x}) + \sum_j \lambda_j \nabla \boldsymbol{g_j}(\boldsymbol{x}) - \boldsymbol{\gamma} = \boldsymbol{0}$, where $\gamma_i = \begin{cases} \alpha_i \text{ if } x_i = a_i \\ -\beta_i \text{ if } x_i = b_i \\ 0 \text{ if } a_i < x_i < b_i \end{cases}$

Since we disallow degeneracy and the variables indexed in D are all strictly between their respective bounds this implies that

$\nabla_I \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{J}_I^T(\boldsymbol{x})\boldsymbol{\lambda} - \boldsymbol{\gamma}_I = \boldsymbol{0}$, and

$\nabla_D \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{J}_D^T(\boldsymbol{x})\boldsymbol{\lambda} = \boldsymbol{0}$.

The second equation yields $\lambda = -[\, \boldsymbol{J}_D^T(\boldsymbol{x})]^{-1}\nabla_D \boldsymbol{f}(\boldsymbol{x})$

Substituting this value for λ into the first equation yields

$$\gamma_I = \nabla_I \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{J}_I^T(\boldsymbol{x})\,[\boldsymbol{J}_D^T(\boldsymbol{x})]^{-1}\nabla_D \boldsymbol{f}(\boldsymbol{x})$$

$$= \nabla_I \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{J}_I^T(\boldsymbol{x})\,[\,\boldsymbol{J}_D^{-1}(\boldsymbol{x})]^T\nabla_D \boldsymbol{f}(\boldsymbol{x}) \qquad (\text{since } (\boldsymbol{A}^T)^{-1}=(\boldsymbol{A}^{-1})^T\ldots)$$

$$= \nabla_I \boldsymbol{f}(\boldsymbol{x}) - \left[[\boldsymbol{J}_D(\boldsymbol{x})]^{-1}\,\boldsymbol{J}_I(\boldsymbol{x})\right]^T\nabla_D \boldsymbol{f}(\boldsymbol{x}) \qquad (\text{since } \boldsymbol{A}^T\boldsymbol{B}^T=(\boldsymbol{B}\boldsymbol{A})^T\ldots)$$

$$= \nabla_I \boldsymbol{f}(\boldsymbol{x}) - \left[\nabla_D \boldsymbol{f}^T(\boldsymbol{x})\,[\boldsymbol{J}_D(\boldsymbol{x})]^{-1}\,\boldsymbol{J}_I(\boldsymbol{x})\right]^T \qquad (\text{since } \boldsymbol{A}^T\boldsymbol{B}=(\boldsymbol{B}^T\boldsymbol{A})^T\ldots)$$

Note that this is the same formula we obtained for the reduced gradient vector! Thus the method may be viewed as moving through a sequence of feasible points $\boldsymbol{x}^k$ "in search of" a point that yields a λ vector that satisfies the K-K-T conditions for optimality. This search is done using the above relationship.

EXAMPLE:

$$\text{Min} \quad x_1^2 - x_1 - x_2$$

$$\text{st} \quad 2x_1 + x_2 \le 1,$$

$$x_1 + 2x_2 \le 1, \quad x_1, x_2 \ge 0$$

Rewriting this in the standard form required for GRG

$$\text{Min} \quad f(\boldsymbol{x}) = x_1^2 - x_1 - x_2$$

$$\text{st} \quad g_1(\boldsymbol{x}) = 2x_1 + x_2 + x_3 \quad -1=0$$

$$g_2(\boldsymbol{x}) = x_1 + 2x_2 \quad + x_4 -1=0$$

$$(a_i=)\ 0 \le x_i \le 1\ (=b_i)$$

Note that we could have actually used (½) for b_1 and b_2 in this problem. In cases where the bound is not obvious from the constraints we could use some large value.

Let us start with

$\boldsymbol{x^0} = [x_1^0 \quad x_2^0 \quad x_3^0 \quad x_4^0]^T = [¼ \quad 0 \quad ½ \quad ¾]^T$

with $f(\boldsymbol{x^0}) = -3/16$.

Let us (arbitrarily) choose x_1 and x_2 as the independent variables.

The requirements are that the point should be feasible with respect to both the constraints and the bounds (which is obviously met), the dependent variables cannot be at either bound (also met), and finally that $\boldsymbol{J}_D$ should be nonsingular (also met since $\boldsymbol{J}_D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, which is obviously nonsingular).

ITERATION 1: $\boldsymbol{x}_{\mathrm{I}} = \begin{bmatrix} 1/4 \\ 0 \end{bmatrix}, \quad \boldsymbol{x}_{\mathrm{D}} = \begin{bmatrix} 1/2 \\ 3/4 \end{bmatrix}$

$$\boldsymbol{\nabla f}(\boldsymbol{x}) \begin{bmatrix} 2x_1 - 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}; \qquad \boldsymbol{J}(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \frac{\partial g_1}{\partial x_3} & \frac{\partial g_1}{\partial x_4} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \frac{\partial g_2}{\partial x_3} & \frac{\partial g_2}{\partial x_4} \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{bmatrix}$$

Thus $\boldsymbol{\nabla}_{\mathrm{D}} \boldsymbol{f}(\boldsymbol{x^0}) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $\qquad \boldsymbol{\nabla}_{\mathrm{I}} \boldsymbol{f}(\boldsymbol{x^0}) = \begin{bmatrix} -1/2 \\ -1 \end{bmatrix}$

and $\boldsymbol{J}_{\mathrm{D}}(\boldsymbol{x^0}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $\qquad \boldsymbol{J}_{\mathrm{I}}(\boldsymbol{x^0}) = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$

Compute the reduced gradient...

$$\gamma_{\mathrm{I}} = \nabla_{\mathrm{I}} \boldsymbol{f}(\boldsymbol{x^0}) \; - \; \left[\nabla_{\mathrm{D}} \boldsymbol{f}^{\mathrm{T}}(\boldsymbol{x^0}) \, [\boldsymbol{J}_{\mathrm{D}}(\boldsymbol{x^0})]^{-1} [\boldsymbol{J}_{\mathrm{I}}(\boldsymbol{x^0})]\right]^{\mathrm{T}}$$

(NOTE: A corresponding step in the Simplex method is the computation of the "$c_j - z_j$" values; the reduced costs.)

$$\therefore \qquad \boldsymbol{\gamma}_{\mathrm{I}} = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} = \begin{bmatrix} -1/2 \\ -1 \end{bmatrix} - \left\{ [0 \;\; 0] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \right\}^{T} = \begin{bmatrix} -1/2 \\ -1 \end{bmatrix}$$

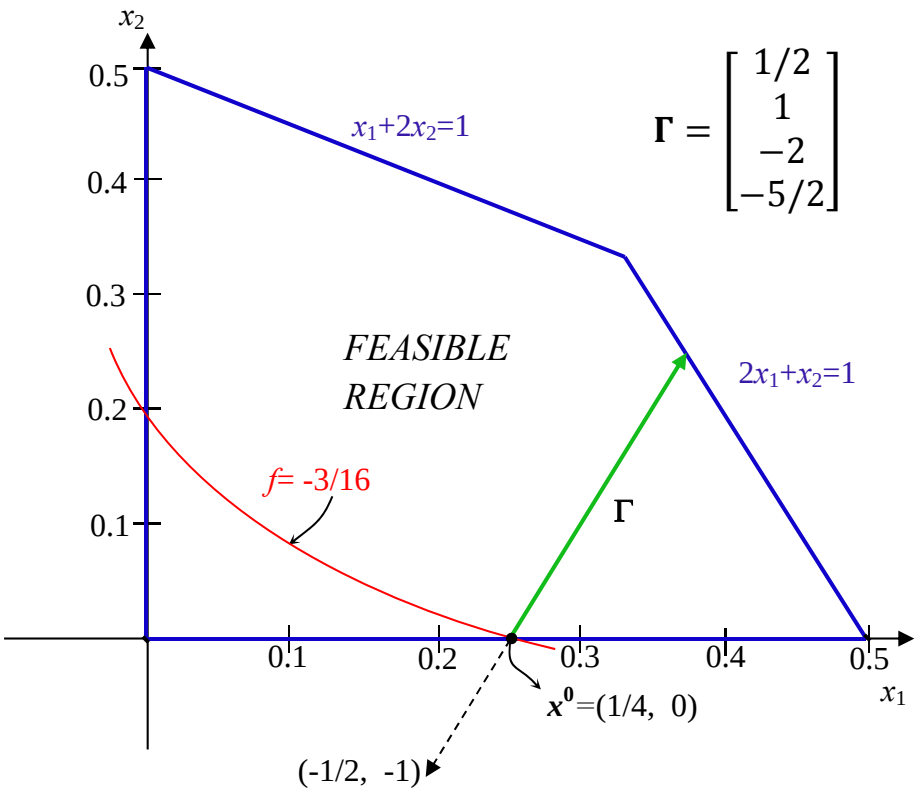
This says that the objective function f increases in the direction $[-1/2 \quad -1]^{\mathrm{T}}$.

Therefore our search should proceed in the opposite direction, i.e.,

$$\boldsymbol{\Gamma}_{\mathrm{I}} = -\boldsymbol{\gamma}_{\mathrm{I}} = \begin{bmatrix} 1/2 \\ 1 \end{bmatrix} = \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix}$$

Note that Γ_2=1 is possible only because x_2 is currently at its lower bound of a_2 =0. If it had been at its upper bound b_2 we would have chosen Γ_2=0.

$$\therefore \quad \boldsymbol{\Gamma}_{\mathrm{D}} = -\boldsymbol{J}_{\mathrm{D}}^{-1} \boldsymbol{J}_{\mathrm{I}} \boldsymbol{\Gamma}_{\mathrm{I}} = - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 1/2 \\ 1 \end{bmatrix} = \begin{bmatrix} -2 \\ -5/2 \end{bmatrix} = \begin{bmatrix} \Gamma_3 \\ \Gamma_4 \end{bmatrix}$$



LINE SEARCH: We now minimize along the direction of search:

Minimize $f(\mathbf{x^0}+\alpha\Gamma)$, st $\mathbf{a} \le \mathbf{x^0}+\alpha\Gamma \le \mathbf{b}$

i.e., Minimize $f(\mathbf{x^0}+\alpha\Gamma)$, st $0 \le \alpha \le \alpha_{max}$

where α_{max} = supremum$\{\alpha | \mathbf{a} \le \mathbf{x^0}+\alpha\Gamma \le \mathbf{b}\}$

$$\mathbf{x^0} + \alpha\Gamma = \begin{bmatrix} 1/4 + \alpha/2 \\ 0 + \alpha \\ 1/2 - 2\alpha \\ 3/4 - 5\alpha/2 \end{bmatrix} \Rightarrow \alpha_{max} = \left\{ \alpha \middle| \begin{matrix} 0 - 1/4 \le \alpha(1/2) & \le 1 - 1/4 \\ 0 - 0 \quad \le \alpha(1) & \le 1 - 0 \\ 0 - 1/2 \le \alpha(-2) & \le 1 - 1/2 \\ 0 - 3/4 \le \alpha(-5/2) & \le 1 - 3/4 \end{matrix} \right\}$$

$\Rightarrow$ $-0.5 \le \alpha \le 1.5$; $0 \le \alpha \le 1$; $-0.25 \le \alpha \le 0.25$; $-0.1 \le \alpha \le 0.3$

$\Rightarrow$ α_{max} = Min (1.5, 1, 0.25, 0.3) = **0.25**

Hence the line search is $\underset{0 \le \alpha \le 0.25}{Minimize}\ f \begin{pmatrix} 1/4 + \alpha/2 \\ \alpha \\ 1/2 - 2\alpha \\ 3/4 - 5\alpha/2 \end{pmatrix}$

$$\Rightarrow \underset{0\le\alpha\le 0.25}{Minimize}[(1/4)+(\alpha/2)]^2 - [(1/4)+(\alpha/2)] - \alpha$$

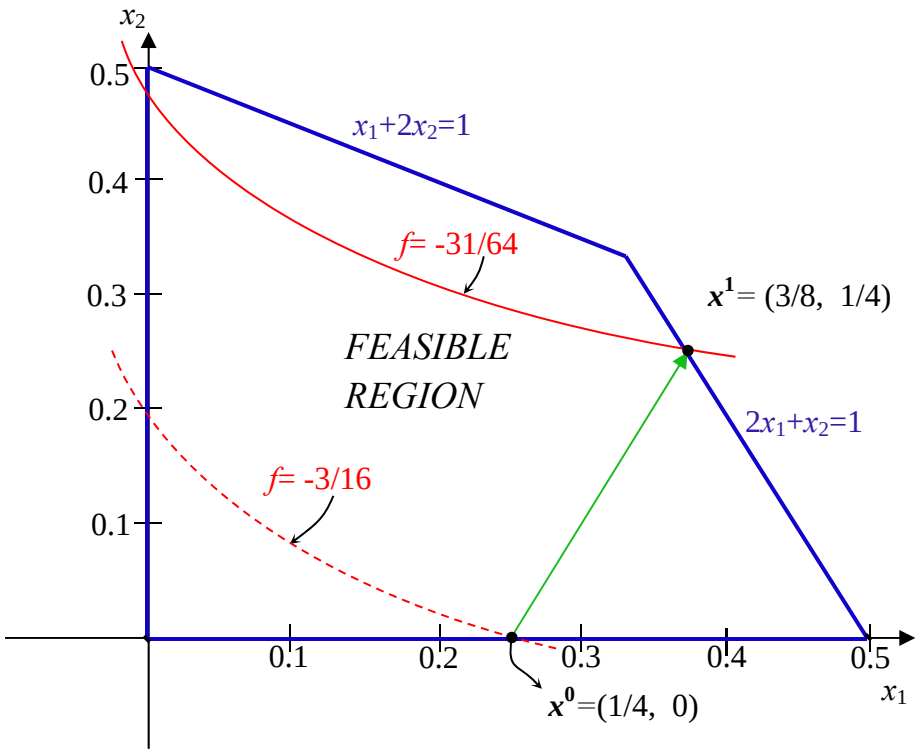
$$\Rightarrow \underset{0\le\alpha\le 0.25}{Minimize}\ \frac{1}{4}\alpha^2 + \frac{5}{4}\alpha - \frac{3}{16}$$

This could be done by a line search procedure such as Fibonacci, GSS, quadratic interpolation, etc. It is easily shown that the optimum is at α^*=0.25. Thus we have:

$$\boldsymbol{x^1} = \boldsymbol{x^0} + \alpha^*\boldsymbol{\Gamma} = \begin{bmatrix}1/4\\0\\1/2\\3/4\end{bmatrix} + \left(\frac{1}{4}\right)\begin{bmatrix}1/2\\1\\-2\\-5/2\end{bmatrix} = \begin{bmatrix}3/8\\1/4\\0\\1/8\end{bmatrix}; \qquad f(\boldsymbol{x^1}) = -31/64$$

Checking for feasibility, we see that $\boldsymbol{x^1}$ is feasible; so we need not make any correction step (in general, this will be true here because we have all linear constraints...)

The current point is shown below:



At the point $\boldsymbol{x}^{\mathbf{1}}$, x_3 has reached its lower bound. Recall that a dependent variable (such as x_3) cannot be at a bound. We must therefore make x_3 independent and remove it from $\boldsymbol{x}_{\mathrm{D}}$ and place it in $\boldsymbol{x}_{\mathrm{I}}$.

The old partition was I={1,2} and D={3,4}.Since neither independent variable is currently at a lower bound, we could choose either x_1 or x_2 to replace, as long as the resulting $\boldsymbol{J}_{\mathrm{D}}$ is. Let us (arbitrarily) choose to replace x_1. Thus D={1,4} and I={2,3}.

ITERATION 2:

$$\boldsymbol{x}^{\mathbf{1}} = [3/8 \quad 1/4 \quad 0 \quad 1/8]^{\mathrm{T}}; \qquad \boldsymbol{x}_{\mathrm{I}} = \begin{bmatrix} x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1/4 \\ 0 \end{bmatrix}, \qquad \boldsymbol{x}_{\mathrm{D}} = \begin{bmatrix} x_1 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3/8 \\ 1/8 \end{bmatrix}$$

$$\boldsymbol{\nabla f}(\boldsymbol{x}) \begin{bmatrix} 2x_1 - 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} \implies \qquad \boldsymbol{\nabla}_{\mathrm{D}}\boldsymbol{f}(\boldsymbol{x}^{\mathbf{1}}) = \begin{bmatrix} -1/4 \\ 0 \end{bmatrix}; \quad \boldsymbol{\nabla}_{\mathrm{I}}\boldsymbol{f}(\boldsymbol{x}^{\mathbf{1}}) = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

$$\boldsymbol{J}(\boldsymbol{x}) = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{bmatrix} \implies \quad \boldsymbol{J}_{\mathrm{D}}(\boldsymbol{x}^{\mathbf{1}}) = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix}; \qquad \boldsymbol{J}_{\mathrm{I}}(\boldsymbol{x}^{\mathbf{1}}) = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix}$$

$$\text{and } \boldsymbol{J}_{\mathrm{D}}^{-1}(\boldsymbol{x}^{\mathbf{1}}) = \begin{bmatrix} 1/2 & 0 \\ -1/2 & 1 \end{bmatrix}$$

Recompute the reduced gradient:

$$\gamma_{\mathrm{I}} = \begin{bmatrix} \gamma_2 \\ \gamma_3 \end{bmatrix} = \nabla_{\mathrm{I}}\boldsymbol{f}(\boldsymbol{x}^{\mathbf{1}}) \; - \; [\nabla_{\mathrm{D}}\boldsymbol{f}^{\mathrm{T}}(\boldsymbol{x}^{\mathbf{1}}) \, [\boldsymbol{J}_{\mathrm{D}}(\boldsymbol{x}^{\mathbf{1}})]^{-1}[\boldsymbol{J}_{\mathrm{I}}(\boldsymbol{x}^{\mathbf{1}})]]^{\mathrm{T}}$$

$$\therefore \qquad \boldsymbol{\gamma}_{\mathrm{I}} = \begin{bmatrix} \gamma_2 \\ \gamma_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \end{bmatrix} - \left\{ [-1/4 \quad 0] \begin{bmatrix} 1/2 & 0 \\ -1/2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix} \right\}^{\mathrm{T}} = \begin{bmatrix} -7/8 \\ 1/8 \end{bmatrix}$$

This says that in order to decrease f, our search direction should be

$$\boldsymbol{\Gamma}_{\mathrm{I}} = -\boldsymbol{\gamma}_{\mathrm{I}} = \begin{bmatrix} 7/8 \\ -1/8 \end{bmatrix} = \begin{bmatrix} \Gamma_2 \\ \Gamma_3 \end{bmatrix}.$$

BUT…

…while x_2=1/4 implies that we can increase it, the fact that x_3=0 implies that we CANNOT decrease it; it is already at its lower bound of a_3=0!

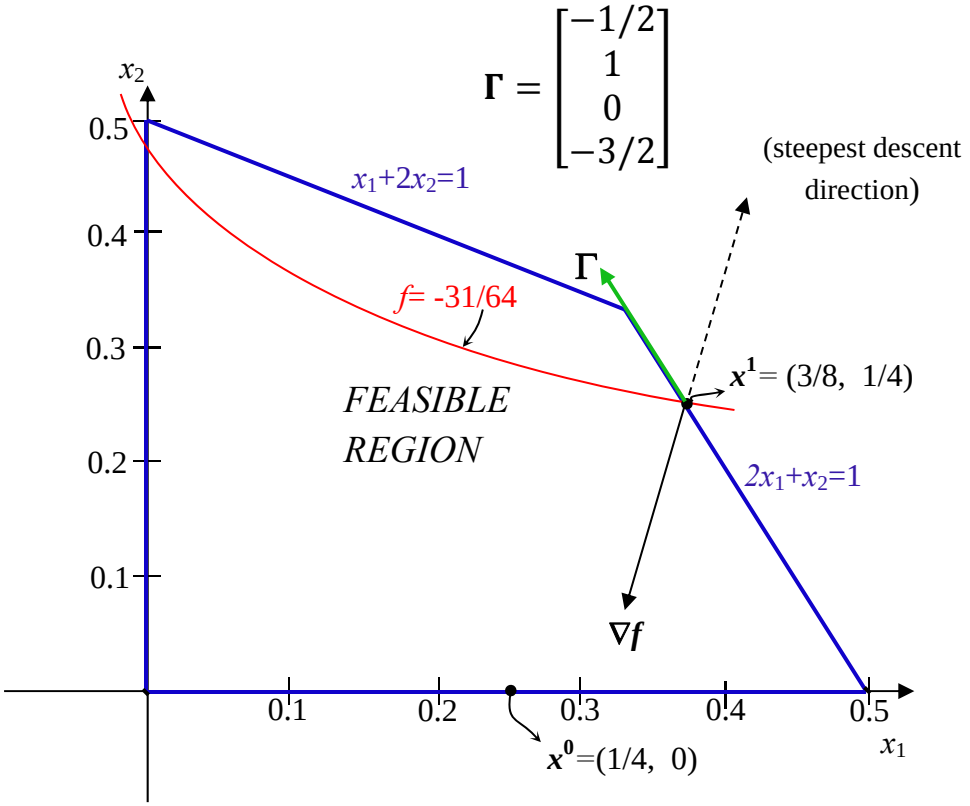
THEREFORE

$\boldsymbol{\Gamma}_{\mathrm{I}} = \begin{bmatrix} 7/8 \\ 0 \end{bmatrix}$, and since we are only looking for a direction, we could simplify matters by choosing $\boldsymbol{\Gamma}_{\mathrm{I}} = \begin{bmatrix} \Gamma_2 \\ \Gamma_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$.

Then

$$\therefore \quad \boldsymbol{\Gamma}_{\mathrm{D}} = \begin{bmatrix} \Gamma_1 \\ \Gamma_4 \end{bmatrix} = -\boldsymbol{J}_{\mathrm{D}}^{-1}\boldsymbol{J}_{\mathrm{I}}\boldsymbol{\Gamma}_{\mathrm{I}} = -\begin{bmatrix} 1/2 & 0 \\ -1/2 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1/2 \\ -3/2 \end{bmatrix}$$

LINE SEARCH: We now minimize along the direction of search:



Minimize $f(\boldsymbol{x}^1+\alpha\boldsymbol{\Gamma})$, st $\boldsymbol{a} \le \boldsymbol{x}^1+\alpha\boldsymbol{\Gamma} \le \boldsymbol{b}$

i.e., Minimize $f(\boldsymbol{x}^1+\alpha\boldsymbol{\Gamma})$, st $0 \le \alpha \le \alpha_{max}$

where α_{max} = supremum$\{\alpha | \ \boldsymbol{a} \le \boldsymbol{x}^1+\alpha\boldsymbol{\Gamma} \le \boldsymbol{b}\}$

$$\boldsymbol{x}^1 = \begin{bmatrix} 3/8 - \alpha/2 \\ 1/4 + \alpha \\ 0 + 0\alpha \\ 1/8 - 3\alpha/2 \end{bmatrix} \Longrightarrow \alpha_{max} = \left\{ \alpha \middle| \begin{matrix} 0 - 3/8 \le \alpha(-1/2) & \le 1 - 3/8 \\ 0 - 1/4 \le \alpha(1) & \le 1 - 1/4 \\ 0 \le \alpha(0) & \le 1 - 0 \\ 0 - 1/8 \le \alpha(-3/2) & \le 1 - 1/8 \end{matrix} \right\}$$

$\Rightarrow$ $\alpha \le 3/4$; $\alpha \le 3/4$; ––; $\alpha \le 1/12$

$\Rightarrow$ α_{max} = Min (3/4,3/4,-,1/12) = 1/12

Hence the line search is $\underset{0 \le \alpha \le 1/12}{Minimize}\ f \begin{pmatrix} 3/8 - \alpha/2 \\ 1/4 + \alpha \\ 0 \\ 1/8 - 3\alpha/2 \end{pmatrix}$

$$\Longrightarrow \underset{0 \le \alpha \le 1/12}{Minimize}\ \frac{1}{4}\alpha^2 - \frac{1}{8}\alpha - \frac{31}{64}$$

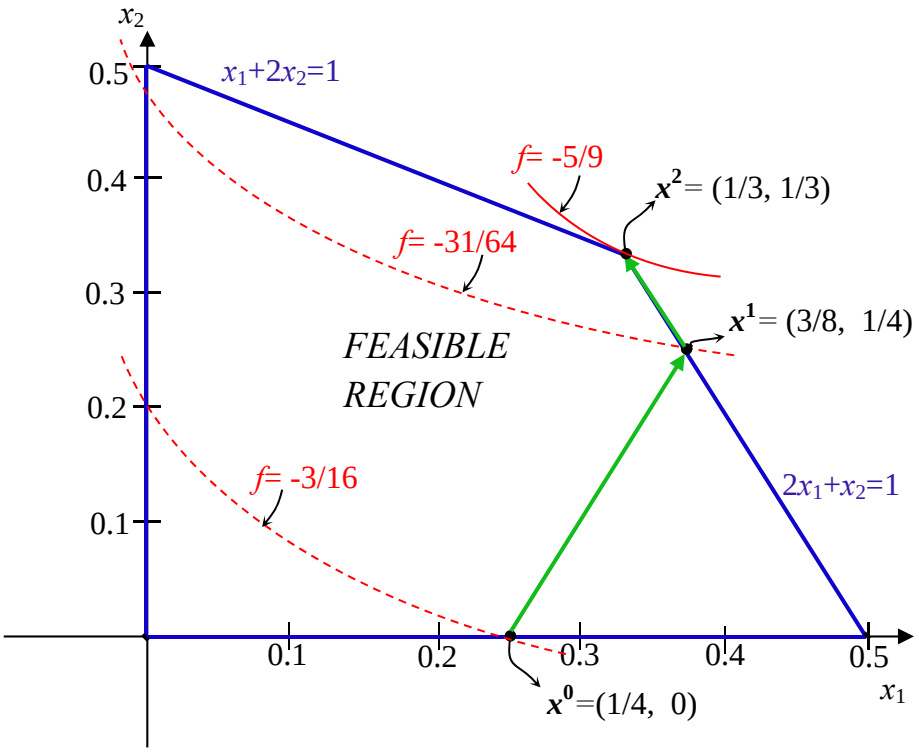
$\Rightarrow \alpha^* = 1/12$. (VERIFY…).

Thus $\boldsymbol{x}^2 = \boldsymbol{x}^1 + \alpha^*\boldsymbol{\Gamma} = \begin{bmatrix} 3/8 \\ 1/4 \\ 0 \\ 1/8 \end{bmatrix} + \left(\frac{1}{12}\right)\begin{bmatrix} -1/2 \\ 1 \\ 0 \\ -3/2 \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 0 \\ 0 \end{bmatrix}$; $f(\boldsymbol{x}^2) = -5/9$

Checking for feasibility, we see that $\boldsymbol{x}^2$ is feasible; so we need not make any correction step...

Again, note that a <u>dependent</u> variable (x_4) has reached its (lower) bound of $a_4 = 0$.

Therefore we must once again repartition...



The old partition was I={1,4} and D={2,3}. We must choose either x_3 or x_2 to replace x_4. However x_3 is already at its (lower) bound so it cannot be made dependent. Therefore, we replace x_2 by x_4.

Our new partition is D={1,2} and I={3,4}

ITERATION 3:

$$\boldsymbol{x^1} = [1/3 \quad 1/3 \quad 0 \quad 0]^{\mathrm{T}}; \qquad \boldsymbol{x}_{\mathrm{I}} = \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \qquad \boldsymbol{x}_{\mathrm{D}} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \end{bmatrix}$$

$$\boldsymbol{\nabla f(x)} \begin{bmatrix} 2x_1 - 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \qquad \boldsymbol{\nabla}_{\mathrm{D}} \boldsymbol{f}(\boldsymbol{x^2}) = \begin{bmatrix} -1/3 \\ -1 \end{bmatrix}; \quad \boldsymbol{\nabla}_{\mathrm{I}} \boldsymbol{f}(\boldsymbol{x^2}) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\boldsymbol{J(x)} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{bmatrix} \Rightarrow \quad \boldsymbol{J}_{\mathrm{D}}(\boldsymbol{x^2}) = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}; \qquad \boldsymbol{J}_{\mathrm{I}}(\boldsymbol{x^2}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\text{and } \boldsymbol{J}_{\mathrm{D}}^{-1}(\boldsymbol{x^2}) = \begin{bmatrix} 2/3 & -1/3 \\ -1/3 & 2/3 \end{bmatrix}$$

Recompute the reduced gradient:

$$\gamma_I = \begin{bmatrix}\gamma_3\\ \gamma_4\end{bmatrix} = \nabla_I \boldsymbol{f}(\boldsymbol{x}^2) \; - \; [\nabla_D \boldsymbol{f}^T(\boldsymbol{x}^2)\,[\boldsymbol{J}_D(\boldsymbol{x}^2)]^{-1}[\boldsymbol{J}_I(\boldsymbol{x}^2)]]^T$$

$$\therefore \qquad \boldsymbol{\gamma}_I = \begin{bmatrix}\gamma_3\\ \gamma_4\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix} - \left\{[-1/3 \quad 1]\begin{bmatrix}2/3 & -1/3\\ -1/3 & 2/3\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\right\}^T = \begin{bmatrix}-1/9\\ 5/9\end{bmatrix}$$

Thus, to decrease *f*, we must move in the direction $-\gamma_I$ and

(a) Increase x_3:this is OK since x_3=0 currently

(b) Decrease x_4: this is <u>not allowed</u> because $x_4 = a_4 = 0$

Hence

$$\boldsymbol{\Gamma}_I = \begin{bmatrix}\Gamma_3\\ \Gamma_4\end{bmatrix} = \begin{bmatrix}1/9\\ 0\end{bmatrix} \equiv \begin{bmatrix}1\\0\end{bmatrix}.$$

and

$$\boldsymbol{\Gamma}_D = \begin{bmatrix}\Gamma_1\\ \Gamma_2\end{bmatrix} = -\boldsymbol{J}_D^{-1}\boldsymbol{J}_I\boldsymbol{\Gamma}_I = -\begin{bmatrix}2/3 & -1/3\\ -1/3 & 2/3\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}-2/3\\ 1/3\end{bmatrix}$$

CONTINUE AND VERIFY THAT YOU GET THE SAME OPTIMAL SOLUTION OBTAINED BY THE **CONVEX SIMPLEX METHOD**...

SEPARABLE PROGRAMMING

DEFINITION: A function $f(x_1,x_2,...,x_n)$ is SEPARABLE if it can be written as a sum of terms, with each term being a function of a SINGLE variable:

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^{n} f_i(x_i)$$

EXAMPLES

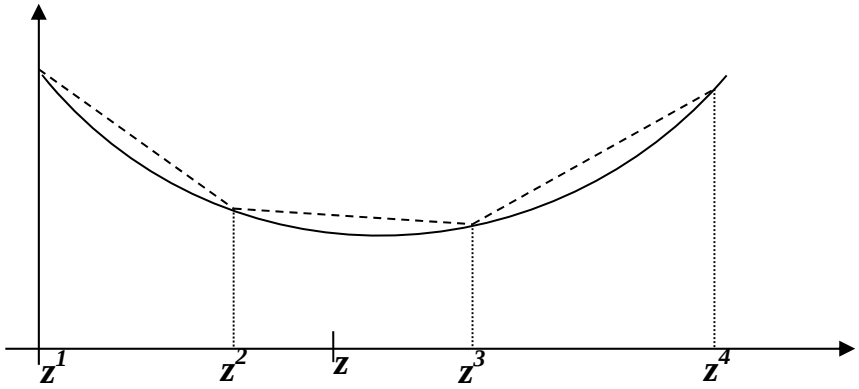
NOT SEPARABLE	SEPARABLE
$x_1 x_2 + x_3$	$\sqrt{x_1} + 2 \log x_2$
$5x_1/x_2 - x_1$	$x_1^2 + 3x_1 + 6x_2 - x_2^2$

$\log(xy^2)$, although this could be separated as $\log x + 2(\log y)$…

PIECE-WISE LINEAR (SEPARABLE) PROGRAMMING

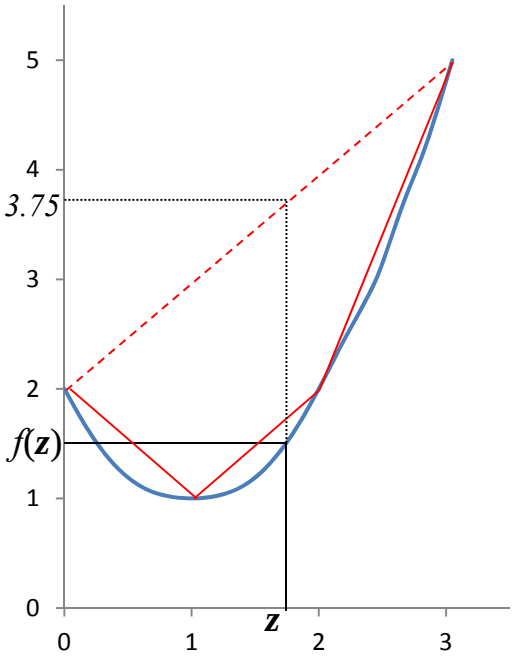
Let $\lambda_1, \lambda_2, \dots$ be "weights" such that $\Sigma_j \lambda_j = 1$, $\lambda_j \geq 0$ for all j. If we have a convex function $f(\mathbf{z})$, we have for any $\mathbf{z}$ given by $\mathbf{z} = \lambda_1 \mathbf{z}^1 + \lambda_2 \mathbf{z}^2 + ... + \lambda_k \mathbf{z}^k$,

$f(\mathbf{z}) \cong \lambda_1 f(\mathbf{z}^1) + \lambda_2 f(\mathbf{z}^2) + ... + \lambda_k f(\mathbf{z}^k)$

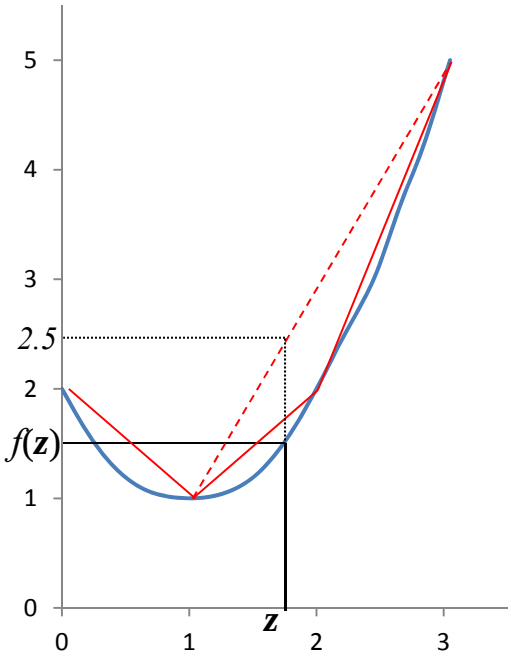


each $\mathbf{z}^i$ is a GRID POINT

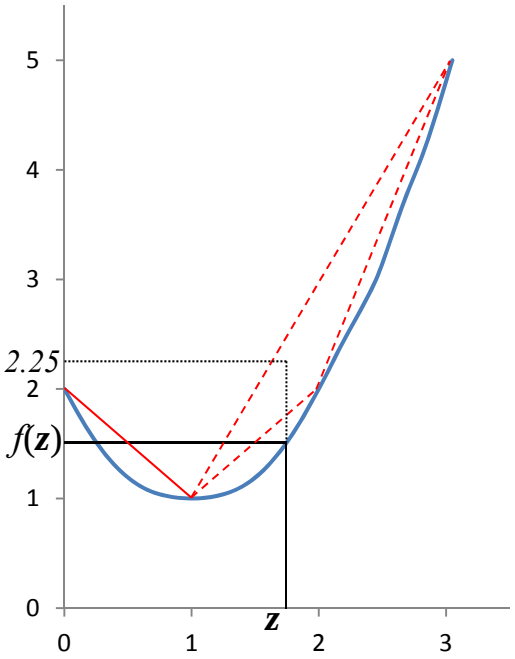
In general, any $\boldsymbol{z}$ has many representations in terms of the $\boldsymbol{z}^k$. Consider $\boldsymbol{z}$=1.75 = 7/4, with $f(\boldsymbol{z})$=1.5, and four grid points $\boldsymbol{z}^0$=0, $f(\boldsymbol{z}^0)$=2; $\boldsymbol{z}^1$ =1, $f(\boldsymbol{z}^1)$=1; $\boldsymbol{z}^2$ =2, $f(\boldsymbol{z}^2)$=2; and $\boldsymbol{z}^3$ =3, $f(\boldsymbol{z}^3)$=5:



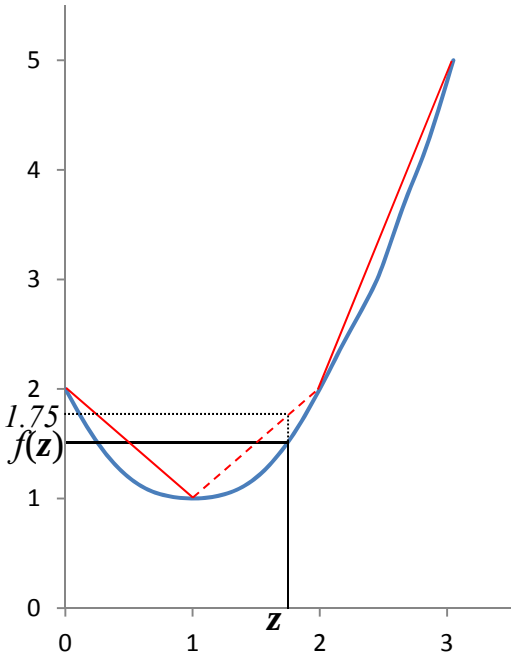
λ_0=5/12, λ_3=7/12, λ_1=λ_2=0;
$f(\boldsymbol{z})$ ≅ 3.75



λ_1=5/8, λ_4=7/12, λ_1=λ_3=0;
$f(\boldsymbol{z})$ ≅ 2.5



λ_1=1/2, λ_2=1/4, λ_3=1/4, λ_0=0;
$f(\boldsymbol{z})$ ≅ 2.25



λ_1=1/4, λ_2=3/4, λ_0=λ_3=0;
$f(\boldsymbol{z})$ ≅ 1.75

Notice that in all cases the approximation to $f(\mathbf{z})$ is always greater than the true value of $f(\mathbf{z})$ (=1.5). However, the specific combination of grid points that yields the lowest value for the (and hence the best) approximation is the one which uses only the two ADJACENT grid points.

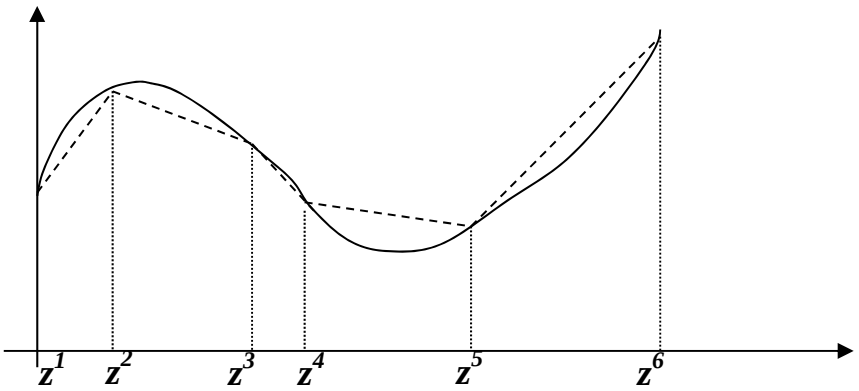
The problem Min $f(\mathbf{z})$, st $g(\mathbf{z})\leq 0$ could now be approximated as

$$\text{Min} \quad \Sigma_k \lambda_k f(\mathbf{z}^k)$$

$$\text{st} \quad \Sigma_k \lambda_k g(\mathbf{z}^k)\leq 0; \quad \Sigma_k \lambda_k = 1, \ \lambda_k \geq 0 \ \forall \ k.$$

This is an LP Problem in the λ variables! Furthermore, in the optimal set of λ values, AT MOST two λ_j values will be positive and these will be ADJACENT.

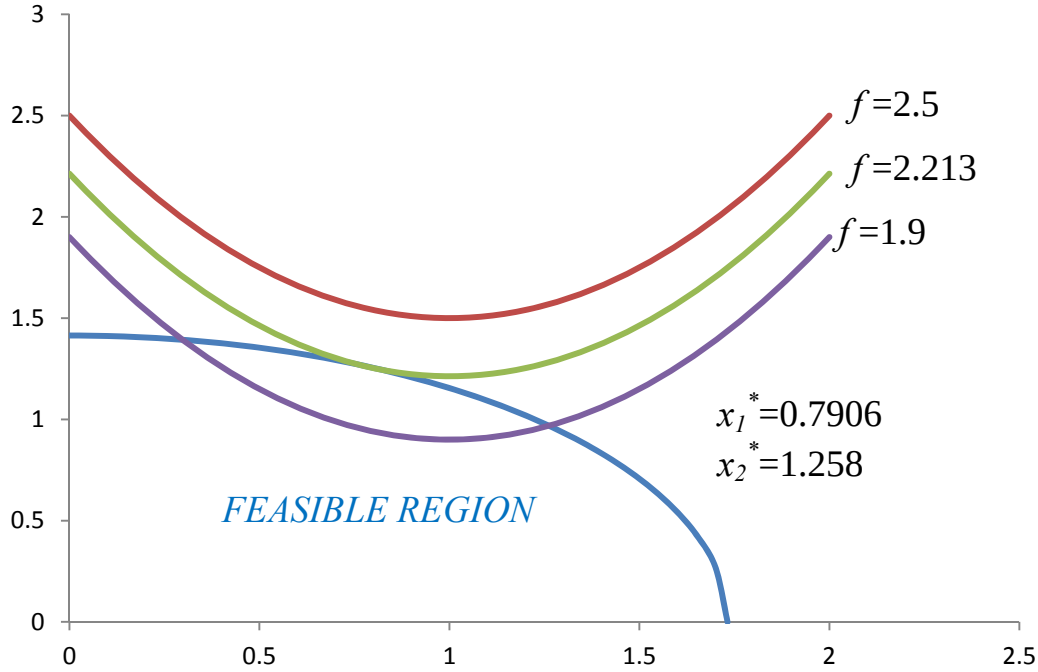
Question: What if $f(\mathbf{z})$ is NOT convex?



In that case, the property above will not necessarily hold; a "RESTRICTED BASIS ENTRY" rule must be enforced to "force" adjacent points in order to get a good approximation.

EXAMPLE Maximize $2x_1 - x_1^2 + x_2$

st $\quad 2x_1^2 + 3x_2^2 \le 6; \quad x_1, x_2 \ge 0.$



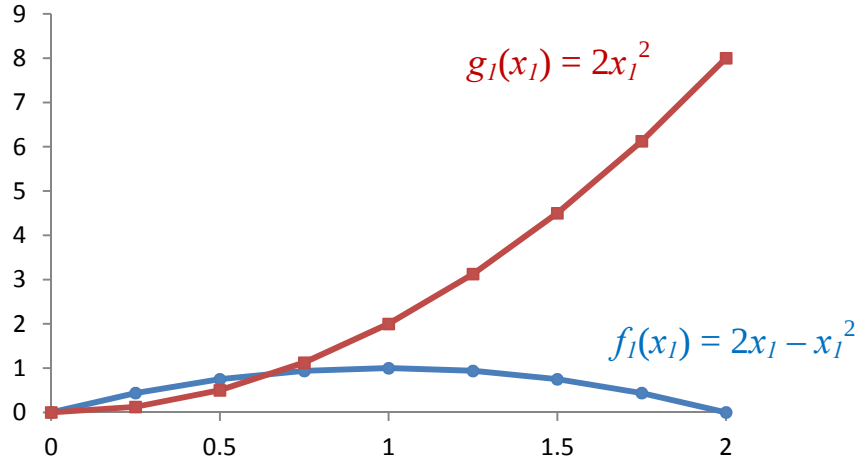
Thus the problem is

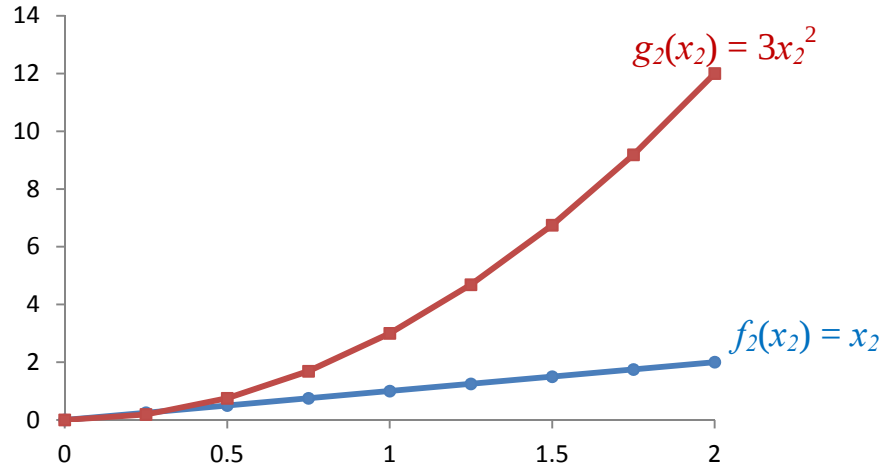
Max $f_1(x_1) + f_2(x_2) = \{2x_1 - x_1^2\} + \{x_2\}$

st $\quad g_1(x_1) + g_2(x_2) = \{2x_1^2\} + \{3x_2^2\} \le 6$

Suppose we use the 9 grid points (0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0) for approximating $f_1$and g_1; and the same 9 grid points for approximating f_2 and g_2.

k	x_1	f_1	g_1	k	x_2	f_2	g_2
0	0.00	0.00	0.00	0	0.00	0.00	0.00
1	0.25	0.4375	0.125	1	0.25	0.25	0.1875
2	0.50	0.75	0.50	2	0.50	0.50	0.75
3	0.75	0.9375	1.125	3	0.75	0.75	1.6875
4	1.00	1.00	2.00	4	1.00	1.00	3.00
5	1.25	0.9375	3.125	5	1.25	1.25	4.6875
6	1.50	0.75	4.50	6	1.50	1.50	6.75
7	1.75	0.4375	6.125	7	1.75	1.75	9.1875
8	2.00	0.00	8.00	8	2.00	2.00	12.00





Assuming weights α_k and β_k respectively for the gridpoints corresponding to x_1 and x_2, the PIECE-WISE LINEAR approximation yields the following LP:

Maximize $\sum_{k=0}^{8} f_{1k}\alpha_k + \sum_{k=0}^{8} f_{2k}\beta_k$

st $\sum_{k=0}^{8} g_{1k}\alpha_k + \sum_{k=0}^{8} g_{2k}\beta_k \leq 6$

$\sum_{k=0}^{8} \alpha_k = 1, \quad \sum_{k=0}^{8} \beta_k = 1; \quad \alpha_k, b_k \geq 0$ for all k.

Max $0\alpha_0 + 0.4375\alpha_1 + 0.75\alpha_2 + \ldots + 0.4375\alpha_7 + 0\alpha_8$

$+ 0\beta_0 + 0.25\beta_1 + 0.5\beta_2 + \ldots + 1.75\beta_7 + 2\beta_8$

$\Rightarrow$ st $(0\alpha_0 + 0.125\alpha_1 + \ldots + 8\alpha_8)\ 4 + (0\beta_0 + \ldots + 12\beta_8) \leq 6$

$\alpha_0 + \alpha_1 + \ldots + \alpha_8 = 1, \quad \beta_0 + \beta_1 + \ldots + \beta_8 = 1; \quad \alpha_k, b_k \geq 0$ for all k.

Thus we have a linear program in 18 variables and 3 constraints.

The OPTIMAL SOLUTION to this LP is

$\alpha_3^* = 1$, $\alpha_k^* = 0$ for $k \neq 3$; and

$\beta_5^* = 0.90909...$, $\beta_6^* = 0.09090...$, $\beta_k^* = 0$ for $k \neq 5,6$,

with an objective function value of 2.21023.

Reconstructing the original problem, we have,

$x_1^* = 1(0.75) = 0.75$,

$x_2^* = 0.90909\ (1.25) + 0.09090\ (1.5) = 1.2727$,

with $f(\boldsymbol{x}^*) = 2.21023$,

which compares quite favorably with the true optimum,

$x_1^* = 0.7906$, $x_2^* = 1.258$, with $f(\boldsymbol{x}^*) = 2.213$

The piece-wise LP approach and separable programming are somewhat "specialized" in nature, with limited general applicability. Often, one could try and INDUCE separability...

For example:

Term	Substitution	Extra Constraints	Restrictions
x_1x_2	$x_1x_2 = y_1 - y_2$	$y_1=(x_1+x_2)/2$ $y_2=(x_1- x_2)/2$	--
x_1x_2	$x_1x_2 = y$	$\log y = \log x_1 + \log x_2$	$x_1, x_2 > 0$
$e^{x_1+x_2^2}$	$e^{x_1+x_2^2} = y$	$\log y = (x_1 + x_2^2)$	--

FRACTIONAL PROGRAMMING

The linear fractional programming problem is:

$$\text{Minimize} \quad \frac{\boldsymbol{p}^{\mathrm{T}}\boldsymbol{x} + \alpha}{\boldsymbol{q}^{\mathrm{T}}\boldsymbol{x} + \beta}, \qquad \text{st } \boldsymbol{Ax} = \boldsymbol{b}, \qquad \boldsymbol{x} \geq \boldsymbol{0}$$

This has some interesting properties:

1. If an optimum solution exists, then an extreme point is an optimum (*cf* LP)
2. Every local minimum is also a GLOBAL minimum

Thus we could use a simplex like approach of moving from one extreme point to an adjacent one. The most obvious approach is Zangwill's convex simplex algorithm that we saw earlier, which simplifies here into a minor modification of the Simplex method for LP. The direction finding and step size subproblems are considerably simplified.

Other fractional programming algorithms include the ones developed by Gilmore and Gomory, and Charnes and Cooper.

Refer to Bazaraa, Sherali and Shetty for example, for more details...