INSTITUTO TECNOLÓGICO DE AERONÁUTICA

MP-288

OPTIMIZATION IN MECHANICAL ENGINEERING

Rafael T. L. Ferreira

IEM

2015

NUMERICAL METHODS FOR UNCONSTRAINED DESIGN

Rafael T. L. Ferreira

NUMERICAL METHODS FOR UNCONSTRAINED DESIGN

Reading material (see References at the last slide):

-Chapter 10 of ARORA (2012);

-Chapters 2 and 3 of VANDERPLAATS (2005);

-Chapter 4 of HAFTKA and GÜRDAL (1992).

NUMERICAL METHODS FOR UNCONSTRAINED DESIGN

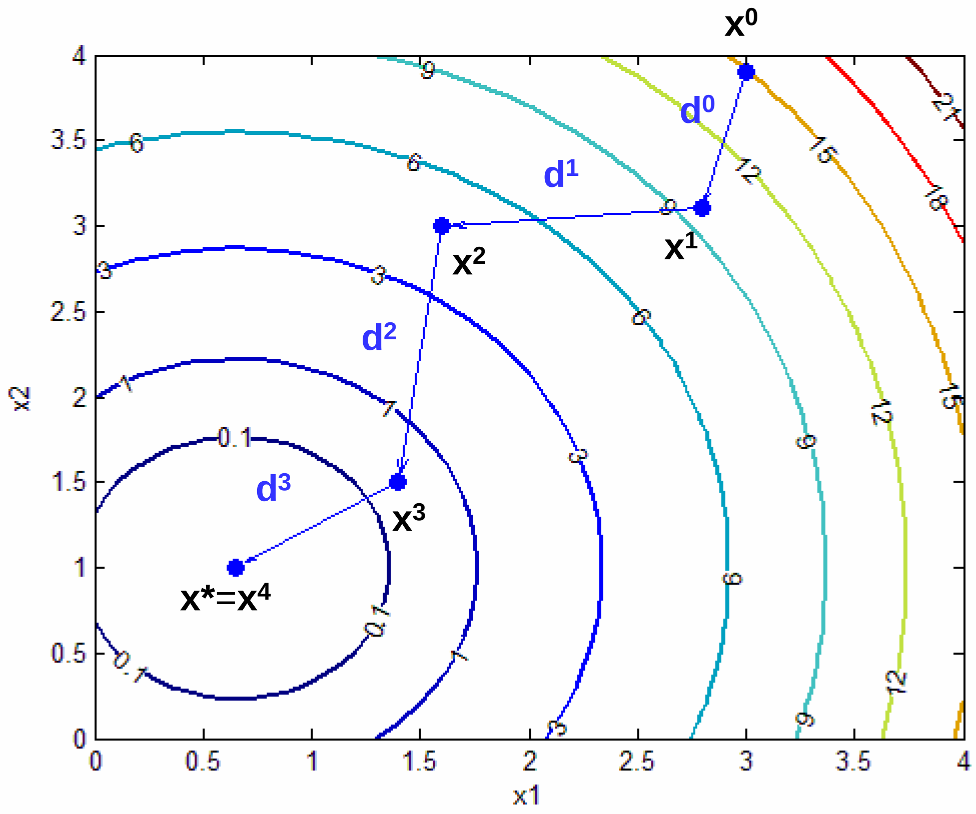
- ÅDIRECTIONAL SEARCH FOR UNCONSTRAINED DESIGN
- ÅLINE SEARCH
- ÅGOLDEN SECTION
- ÅSTEEPEST DESCENT
- ÅCONJUGATE GRADIENT
- ÅINTERIOR PENALTY FUNCTION
- ÅAUGMENTED LAGRANGIAN
- ÅREFERENCES

DIRECTIONAL SEARCH FOR UNCONSTRAINED DESIGN

Minimizing a function of several variables along search directions.

It is an idea largely employed on numerical methods devoted to function minimization.

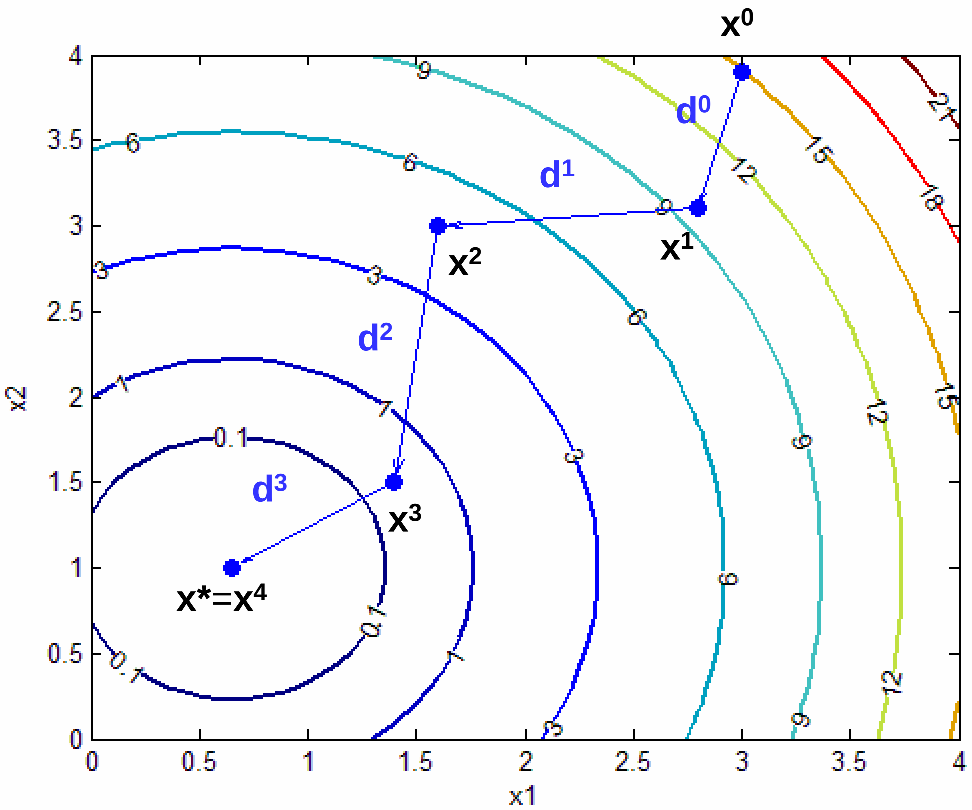
The general procedure is illustrated in the following.



DIRECTIONAL SEARCH FOR UNCONSTRAINED DESIGN

Example: $\min_{x_1,x_2} f(x_1,x_2)$

start at $\mathbf{x}^0$
define direction $\mathbf{d}^0$
minimize $f(\mathbf{x})$ along $\mathbf{d}^0$
find $\mathbf{x}^1$
at $\mathbf{x}^1$
define direction $\mathbf{d}^1$
minimize $f(\mathbf{x})$ along $\mathbf{d}^1$
find $\mathbf{x}^2$
at $\mathbf{x}^2$
...
...
find $\mathbf{x}^*$
stop



LINE SEARCH

Process of finding the minimum of a function of several variables along an established direction **d**.

The assumption of a search direction reduces the number of variables to one in this search.

Methodology:

given $\min_{\mathbf{x}} f(\mathbf{x})$

assume $\mathbf{d}$

assume $\mathbf{x}=\mathbf{x}^k+\alpha\mathbf{d}^k$

such as $f(\mathbf{x})=f(\mathbf{x}^k+\alpha\mathbf{d}^k)$

then now $\min_{\alpha} f(\alpha)$

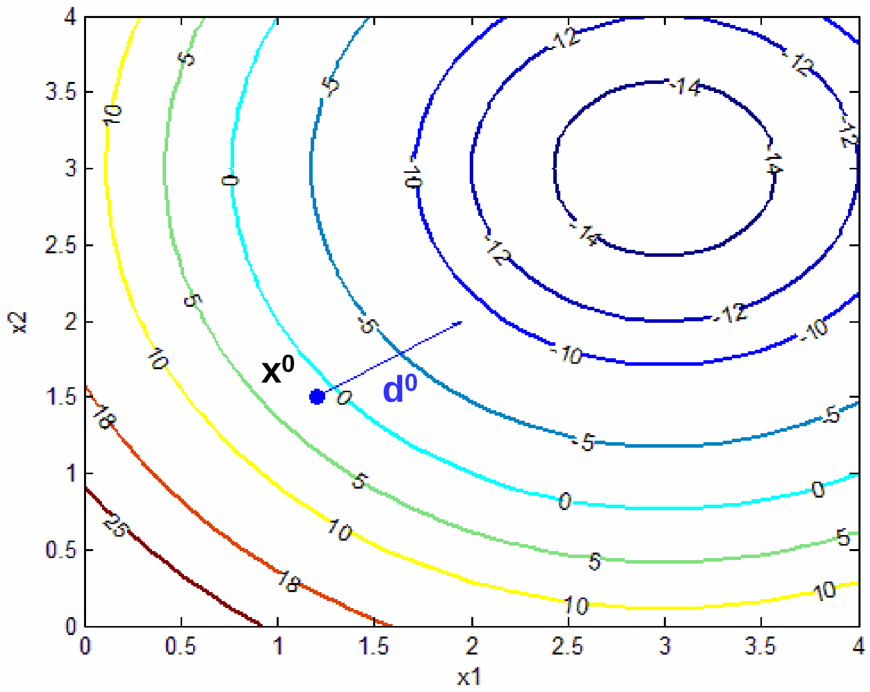
find α^*

$\mathbf{x}^*=\mathbf{x}^{k+1}=\mathbf{x}^k+\alpha^*\mathbf{d}^k$

LINE SEARCH

Example: minimize a function f(**x**) along a direction $\mathbf{d}^0$ starting from point $\mathbf{x}^0$.

$$\min_{\mathbf{x}} f(\mathbf{x})$$



$$f(\mathbf{x}) = f(x_1, x_2) = 3(x_1 - 2)^2 + 3(x_2 - 3)^2 - 6x_1$$

$$\mathbf{x}^0 = \begin{Bmatrix} 1.20 \\ 1.50 \end{Bmatrix}, \quad \mathbf{d}^0 = \begin{Bmatrix} 0.75 \\ 0.50 \end{Bmatrix}$$

$$\mathbf{x} = \mathbf{x}^k + \alpha \mathbf{d}^k$$

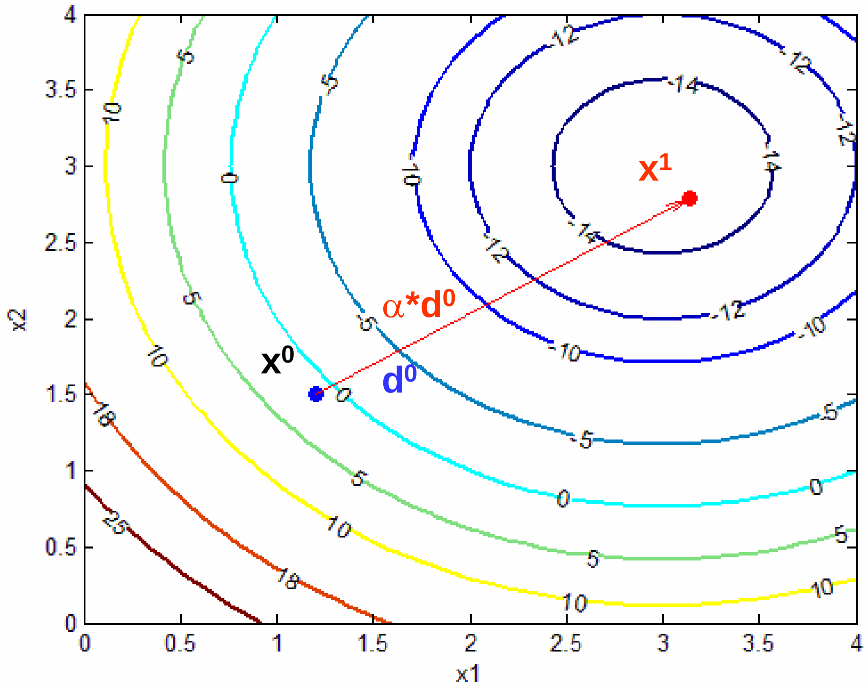
$$\mathbf{x} = \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} 1.20 \\ 1.50 \end{Bmatrix} + \alpha \begin{Bmatrix} 0.75 \\ 0.50 \end{Bmatrix}$$

$$f(\alpha) = 3\left[(1.2 + 0.75\alpha) - 2\right]^2 + 3[(1.5 + 0.5\alpha) - 3]^2 - 6(1.2 + 0.75\alpha)$$

LINE SEARCH

Example: minimize a function f(**x**) along a direction $\mathbf{d}^0$ starting from point $\mathbf{x}^0$.

$$f(\alpha) = 3\left[(1.2 + 0.75\alpha) - 2\right]^2 + 3[(1.5 + 0.5\alpha) - 3]^2 - 6(1.2 + 0.75\alpha)$$



$$\min_{\alpha} f(\alpha)$$

$$\frac{df}{d\alpha} = 0 \implies \alpha^* = 2.585, \quad \left.\frac{d^2 f}{d\alpha^2}\right|_{\alpha=\alpha^*} = 4.875 > 0$$

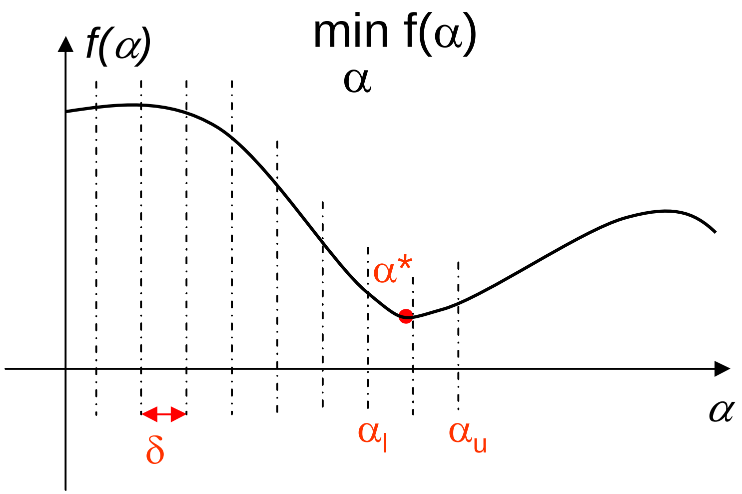
$$\mathbf{x}^* = \mathbf{x}^1 = \mathbf{x}^k + \alpha^* \mathbf{d}^k = \begin{Bmatrix} 3.138 \\ 2.792 \end{Bmatrix}$$

However, in most cases, we do not have an explicit function f(**x**) to optimize!

NUMERICAL LINE SEARCH

Given **d**, numerical determination of α^*.

Strategy: find points α_l and α_u which bound α^*. Reduce this interval iteratively to a predefined tolerance.



Method (Phase I):

- Sample $f(\alpha)$ from $f(0)$ in steps δ, then $\alpha_q = q\delta$;

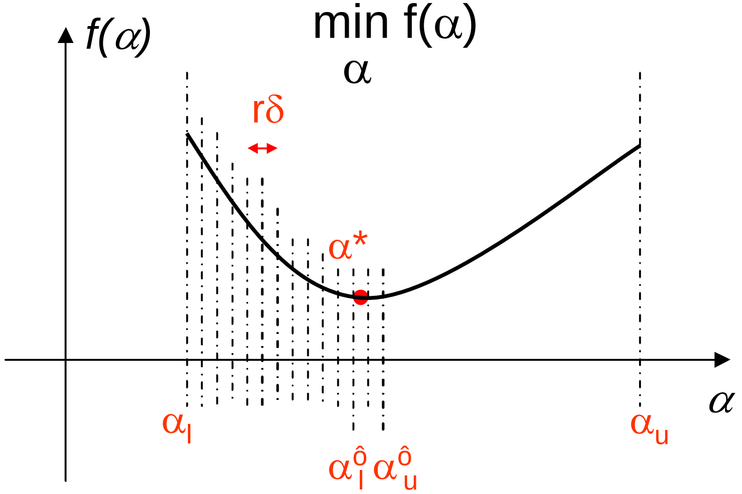
-When $f(\alpha_{q+1}) > f(\alpha_q)$, the minimun is inside $\alpha_l = (q-1)\delta$, $\alpha_u = (q+1)\delta$;

- α^* is bounded.

NUMERICAL LINE SEARCH

Given **d**, numerical determination of α^*.

Strategy: find points α_l and α_u which bound α^*. Reduce this interval iteratively to a predefined tolerance.



Method (Phase II):

-Define the interval of uncertainty $I=\alpha_u-\alpha_l=2\delta$.

-Perform again Phase I, now between α_l and α_u using a new step $\delta^k=r\delta$ $(r<1)$.

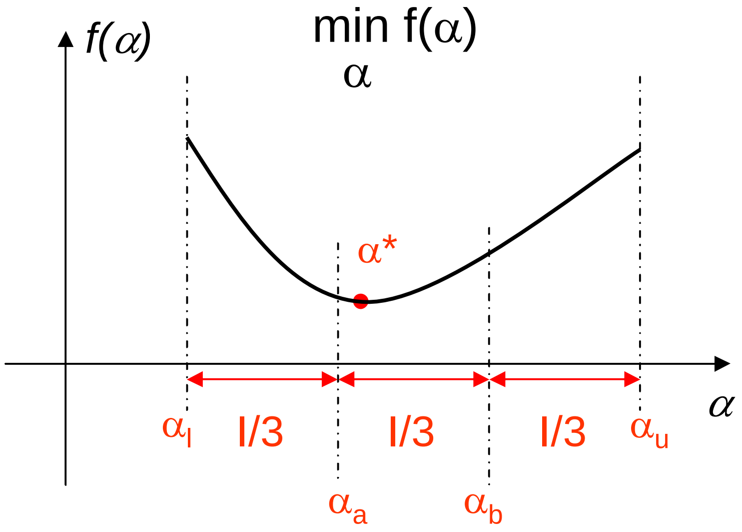
-Repeat the process to reduce the uncertainty to $I<\varepsilon$, $\varepsilon<<1$ (say $\varepsilon=10^{-3}$ to 10^{-6}). (Alternative: use $\delta^k=\varepsilon/2$ as step only one time).

-Assume $\alpha^*=(\alpha_u-\alpha_l)/2$.

NUMERICAL LINE SEARCH

Given **d**, numerical determination of α^*.

Strategy: find points α_l and α_u which bound α^*. Reduce this interval iteratively to a predefined tolerance.



Method (Alternative Phase II):

-Reduce $I=\alpha_u-\alpha_l$ by divinding it in three I/3 equal intervals;

–$\alpha_a=\alpha_l+(1/3)(\alpha_u-\alpha_l)$, $\alpha_b=\alpha_l+(2/3)(\alpha_u-\alpha_l)$;

-If $f(\alpha_a)<f(\alpha_b)$, then $\alpha_u=\alpha_b$ (reduction of I);

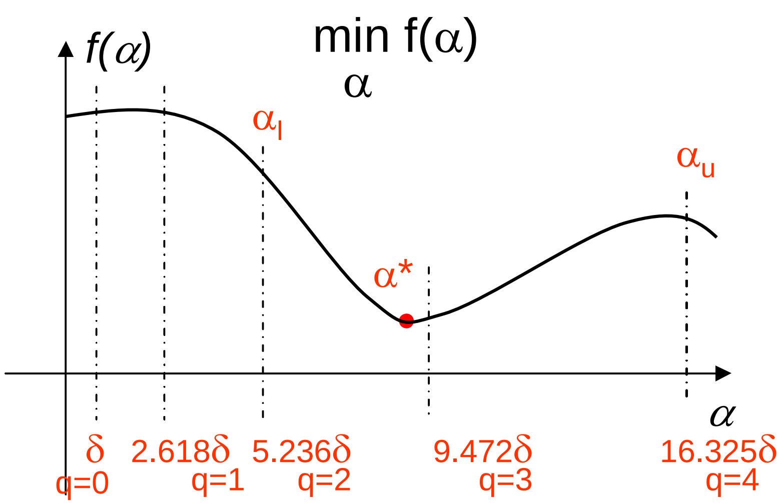
-If $f(\alpha_a)>f(\alpha_b)$, then $\alpha_l=\alpha_a$ (reduction of I);

-Repeat the process with the new bounds α_l,α_u up to $I<\varepsilon$.

-Assume $\alpha^*=(\alpha_u-\alpha_l)/2$.

GOLDEN SECTION METHOD

Efficient variable interval search method. The sampling steps δ increase with the golden ratio $(5^{1/2}+1)/2=1.618...$.

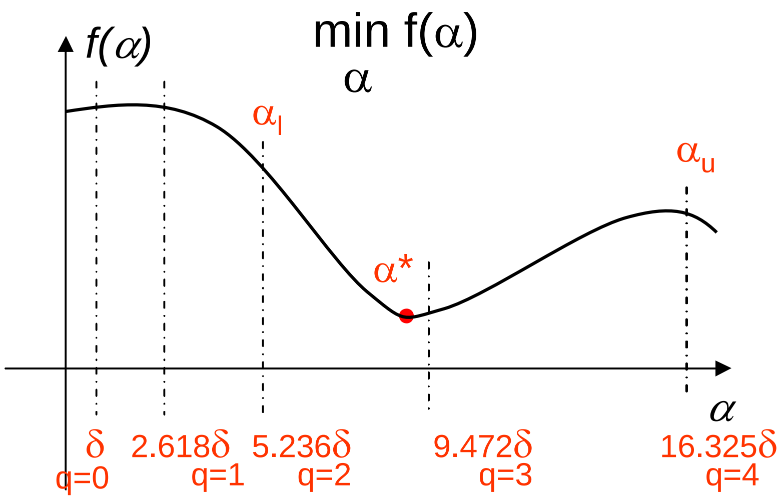


$$\alpha_q = \sum_{j=0}^{q} \delta(1.618)^j, \quad q = 0, 1, 2, \ldots$$

$$q = 0,\ \alpha_0 = \delta$$
$$q = 1,\ \alpha_1 = \delta + \delta 1.618$$
$$q = 2,\ \alpha_2 = \delta + \delta 1.618 + \delta 1.618^2$$
$$q = 3,\ \alpha_3 = \delta + \delta 1.618 + \delta 1.618^2 + \delta 1.618^3$$
$$\ldots$$

GOLDEN SECTION METHOD

Efficient variable interval search method. The sampling steps δ increase with the golden ratio $(5^{1/2}+1)/2=1.618...$.



Method (Phase I):

-Sample $f(\alpha)$ using variable golden ratio steps α_q;

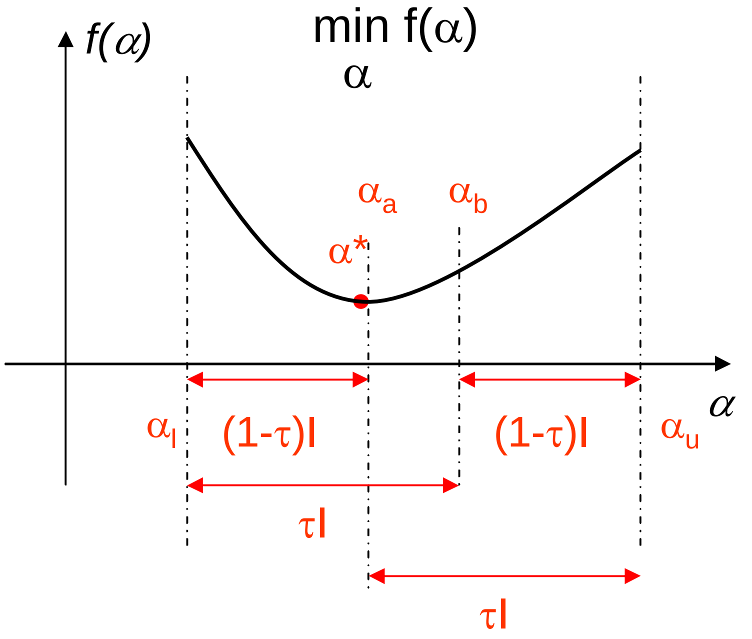
$$\alpha_q = \sum_{j=0}^{q} \delta(1.618)^j, \quad q = 0, 1, 2, \ldots$$

-When $f(\alpha_{q+1})>f(\alpha_q)$, the minimun is inside $\alpha_l=\alpha_{q-1}$, $\alpha_u=\alpha_{q+1}$;

- α^* is bounded.

GOLDEN SECTION METHOD

Efficient variable interval search method. The sampling steps δ increase with the golden ratio $(5^{1/2}+1)/2=1.618...$.



Method (Phase II):

-Reduce the interval of uncertainty, given now by:

$$I = \alpha_u - \alpha_l = \sum_{j=0}^{q+1} \delta(1.618)^j - \sum_{j=0}^{q-1} \delta(1.618)^j = 2.618(1.618^q)\delta$$

- Put two points symetrically located inside the minimum bounds, such that:

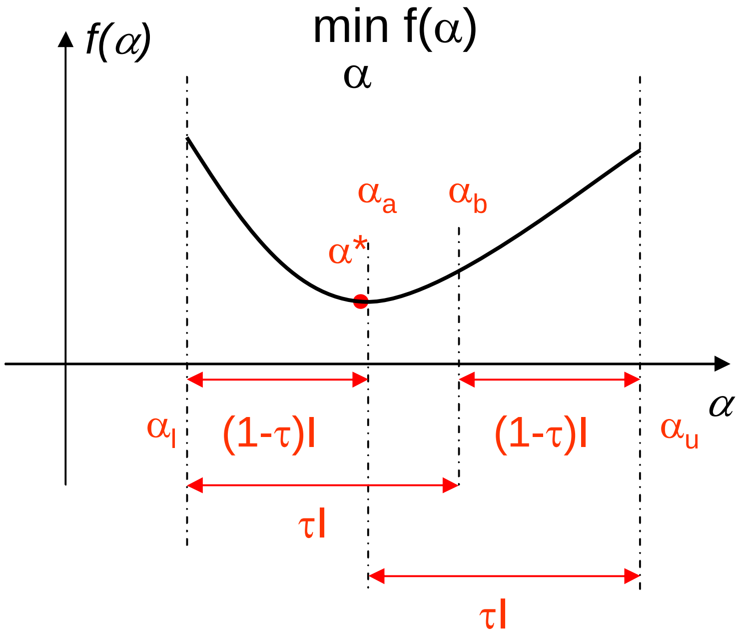
$$\alpha_a = \alpha_l + (1-\tau)(\alpha_u - \alpha_l)$$
$$\alpha_b = \alpha_l + \tau(\alpha_u - \alpha_l)$$

$\tau=1/1.618=0.618$

-Follow next step...

GOLDEN SECTION METHOD

Efficient variable interval search method. The sampling steps δ increase with the golden ratio $(5^{1/2}+1)/2=1.618...$.



Method (Phase II):

-...

-If $f(\alpha_a)<f(\alpha_b)$, then $\alpha_u=\alpha_b$ (reduction of I);

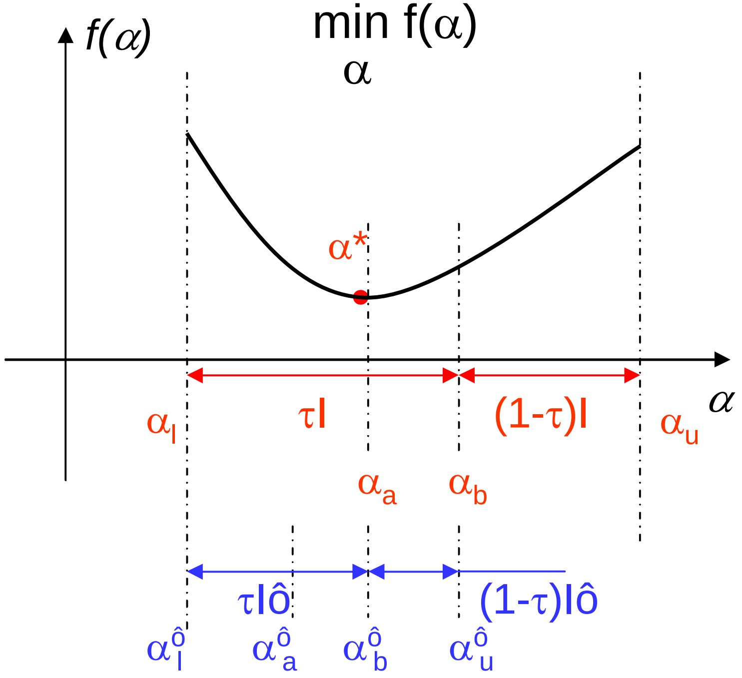
-If $f(\alpha_a)>f(\alpha_b)$, then $\alpha_l=\alpha_a$ (reduction of I);

-Repeat the process with the new bounds α_l,α_u up to $I<\varepsilon$.

-Assume $\alpha^*=(\alpha_u-\alpha_l)/2$.

GOLDEN SECTION METHOD

Efficient variable interval search method. The sampling steps δ increase with the golden ratio $(5^{1/2}+1)/2=1.618...$.



The golden section method reduces the interval of uncertainty keeping a proportion between the old and the new interval.

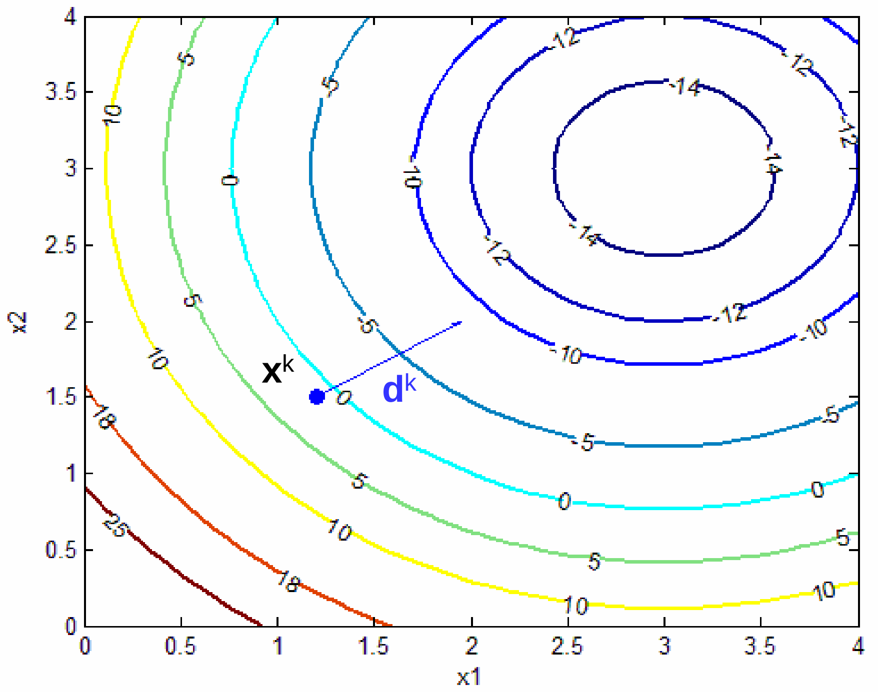
This proportion is such that one of the points α_a or α_b is kept inside the new interval as the new $\alpha\hat{o}_b$ or $\alpha\hat{o}_a$, respectively .

Then, for a new interval Iộ it is necessary only one new function evaluation to carry with the interval reduction.

This makes the golden section method very efficient from a numerical point of view (See Example_Line _Search.xls).

DESCENT SEARCH DIRECTION

Once it is known how to minimize a function over a predefined direction, it is needed to define a descent direction $\mathbf{d}^k$ of the function $f(\mathbf{x})$, at a point $\mathbf{x}^k$, to search in.



$\mathbf{x}^k$

$\mathbf{d}^k$?

STEEPEST DESCENT METHOD

One of the easiest and most efficient ways to find a descent direction to minimize a function.

It adopts the negative of the function gradient as the search direction $\mathbf{d}^k$ at the point $\mathbf{x}^k$.

This is a logical choice, since the gradient of f($\mathbf{x}$) in $\mathbf{x}^k$ points towards to the direction of the fastest increase for f($\mathbf{x}$).

$$\boxed{\mathbf{d}^k = -\nabla f\Big|_{\mathbf{x}^k} \quad d_i = -\frac{\partial f}{\partial x_i}, \; i = 1, \dots, n}$$

STEEPEST DESCENT METHOD

This assumption guarantees minimization. Thinking in the neighborhood of the point $\mathbf{x}^k$, it is valid for a steepest descent direction **d**:

$$\nabla f \cdot \mathbf{d} = \nabla f \cdot (-\nabla f) = -||\nabla f||^2 < 0$$

This result guarantees minimization, since the dot product analyzed is an estimation of f(**x**) in **d** close to $\mathbf{x}^k$.

STEEPEST DESCENT METHOD

Algorithm for the method:

Step 1. Estimate a starting design $\mathbf{x}^0$ and set the iteration counter k=0. Select a convergence parameter $0<\varepsilon<<1$ (say 10^{-5});

Step 2. Calculate the gradient of $f(\mathbf{x})$ at the current point $\mathbf{x}^k$, $\nabla f^k=\nabla f(\mathbf{x}^k)$;

Step 3. Calculate the length of the gradient $||\nabla f^k||$. If $||\nabla f^k||<\varepsilon$, then stop because $\mathbf{x}^k$ is a local minimum. Otherwise, continue;

Step 4. Let the search direction at the current point $\mathbf{x}^k$ be $\mathbf{d}^k=-\nabla f^k$;

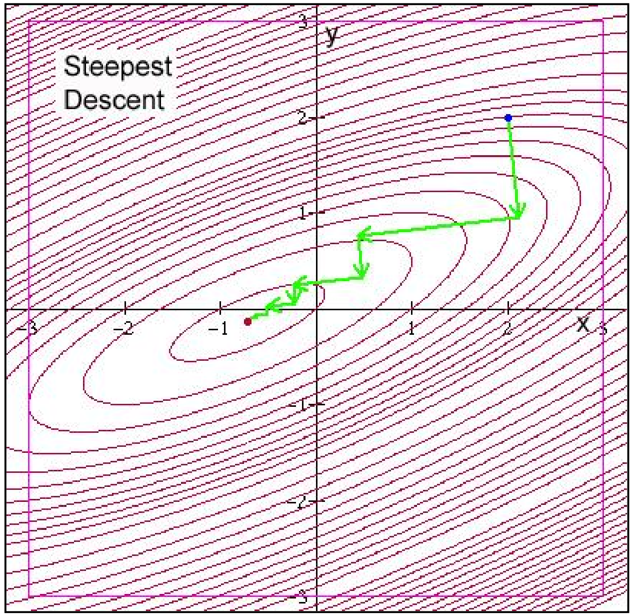
Step 5. Calculate a step size α^{*k} that minimizes $f(\alpha)=f(\mathbf{x}^k+\alpha\mathbf{d}^k)$ (line search), $\mathbf{x}^{k+1}=\mathbf{x}^k+\alpha^{*k}\mathbf{d}^k$. Set k=k+1, go to Step 2.

STEEPEST DESCENT METHOD

Example of function minimization with the steepest descent method.

(Convergence in k=17 iterations.)

$f(x_1,x_2)=x_1^2-3x_1x_2+4x_2^2+x_1-x_2$



initial point $(x_1,x_2)=(2,2)$ $f(2,2)=8$

minimum point $(x_1^*,x_2^*)=(-0.71,-0.14)$ $f(2,2)=3.224$

STEEPEST DESCENT METHOD

There is an interesting resulting concerning with the directions $\mathbf{d}^k$ and $\mathbf{d}^{k+1}$ in the steepest descent method:

The **x** variation is given by:

$$\mathbf{x} = \mathbf{x}^k + \alpha \mathbf{d}^k$$

In line search, we look for a minimum in α where:

$$\frac{df}{d\alpha} = \frac{\partial f}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\alpha} = 0$$

(chain rule)

The point $\mathbf{x}^{k+1}$ is given by:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^* \mathbf{d}^k$$

At the point α^* we then have:

$$\left.\frac{df}{d\alpha}\right|_{\alpha=\alpha^*} = \left.\nabla f\right|_{\mathbf{x}^{k+1}} \cdot \mathbf{d}^k = 0$$

Therefore, the directions $\mathbf{d}^k$ and $\mathbf{d}^{k+1}$ are perpendicular. This slows down the convergence of the method due to zig-zagging.

CONJUGATE GRADIENT METHOD

The conjugate gradient method can be seen as a modified steepest descent method that avoids zig-zagging during function minimization.

It uses as a descent direction $\mathbf{d}^k$:

$$\boxed{\mathbf{d}^k = -\nabla f^k + \beta_k \mathbf{d}^{k-1}, \quad \beta_k = \left(\frac{||\nabla f^k||}{||\nabla f^{k-1}||} \right)^2}$$

In words, it uses as $\mathbf{d}^k$ a combination of the gradient at the point $\mathbf{x}^k$ plus a correction factor multiplying the last used direction $\mathbf{d}^{k-1}$.

Due to FLECHTER and REEVES (1964), this correction is effective in reducing the number of line searches needed to minimize a function.

CONJUGATE GRADIENT METHOD

Algorithm for the method:

Step 1. Estimate a starting design $\mathbf{x}^0$ and set the iteration counter k=0. Select a convergence parameter ε, $0<\varepsilon<<1$ (say 10^{-5}).

Calculate the gradient of f($\mathbf{x}$) at the current point $\mathbf{x}^k$, $\nabla f^k=\nabla f(\mathbf{x}^k)$. If $||\nabla f^k||<\varepsilon$, then stop. Otherwise, assume $\mathbf{d}^0=-\nabla f^0$ and continue to Step 5.

Step 2. Calculate the gradient of f($\mathbf{x}$) at the current point $\mathbf{x}^k$, $\nabla f^k=\nabla f(\mathbf{x}^k)$;

Step 3. Calculate $||\nabla f^k||$. If $||\nabla f^k||<\varepsilon$, then stop. Otherwise, continue;

Step 4. Let the search direction at the current point $\mathbf{x}^k$ be:

$$\mathbf{d}^k = -\nabla f^k + \beta_k \mathbf{d}^{k-1}, \quad \beta_k = \left(\frac{||\nabla f^k||}{||\nabla f^{k-1}||} \right)^2$$

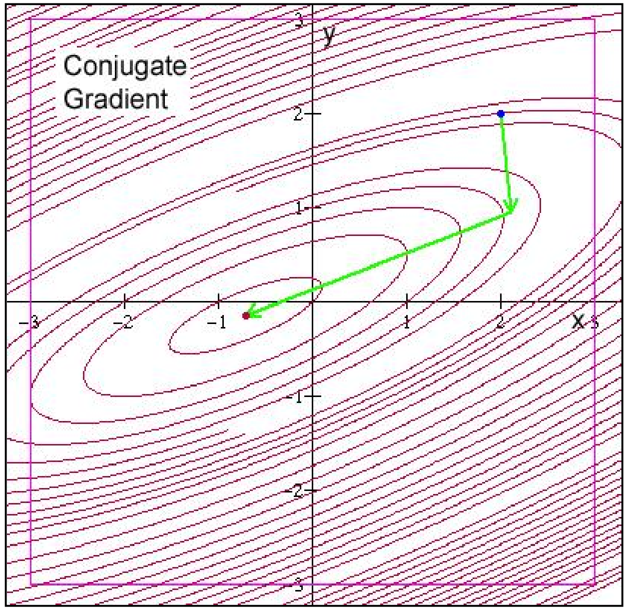
Step 5. Perform line search to α^{*k} that minimizes $f(\alpha)=f(\mathbf{x}^k+\alpha\mathbf{d}^k)$, then $\mathbf{x}^{k+1}=\mathbf{x}^k+\alpha^{*k}\mathbf{d}^k$. Set k=k+1, go to Step 2.

CONJUGATE GRADIENT METHOD

Example of function minimization with the conjugate gradient method.

(Convergence in k=2 iterations, as can be proved to happen to quadratic functions.)

$f(x_1,x_2)=x_1^2-3x_1x_2+4x_2^2+x_1-x_2$



initial point $(x_1,x_2)=(2,2)$ $f(2,2)=8$

minimum point $(x_1^*,x_2^*)=(-0.71,-0.14)$ $f(2,2)=3.224$

CONJUGATE GRADIENT METHOD

Why does it works?

$$\boxed{\mathbf{d}^k = -\nabla f^k + \beta_k \mathbf{d}^{k-1}, \quad \beta_k = \left(\frac{||\nabla f^k||}{||\nabla f^{k-1}||}\right)^2}$$

ñWe know that the gradient vanishes at the optimum.

This means that, if the process is going well, the gradient gets smaller for each iteration.

If this is true, then β_k is a small number and not much correction is applied to the steepest descent direction.

If it is not true, we need more correction (change) in the steepest descent direction and β_k gives precisely that.ò

(Thanks to NIELS OLHOFF, Aalborg University, Denmark.)

CONJUGATE GRADIENT METHOD

The conjugate directions tend to cut diagonally the ortogonal directions of the steepest descent method, saving iterations.

From the algorithm presented, it can be noticed that the first iteration of the conjugate gradient method is exactly a steepest descent iteration.

Both methods converge to the local minimum of f($\mathbf{x}$) nearest to $\mathbf{x}^0$.

REFERENCES

ARORA, J.S. *Introduction to optimum design*, 3rd ed., Elsevier, Amsterdan, 2012.

FLECHTER, R. and REEVES, C.M. Function minimization by conjugate gradients, *The Computer Journal*, v7(2), pp.149-154, 1964. DOI: 10.1093/comjnl/7.2.149

HAFTKA, R.T. and GÜRDAL, Z. *Elements of structural optimization*, 3rd ed., Kluwer, Dordrecht, 1992.

VANDERPLAATS, G.N. *Numerical optimization techniques for engineering design*, 4th ed., Vanderplaats Research and Development, Colorado Springs, 2005.