Minimization of function’s value

outline

- minimization – general consideration
- methods for univariate problems
    - Golden Section
    - Powell interpolation method
- gradient methods for multivariate problems
    - Steepest Descent method
    - Conjugate Gradients method
    - Newton method

Minimization of function

Minimization of function's value is part of wider class of problems commonly called **optimization**. The aim of optimization is to find the set of parameters' values of certain single- or multi-argument mathematical model of: (i) function, (ii) device, (iii) system, etc., which guarantee fulfilling some conditions, the most popular are the minimum or maximum value of certain quantity (energy, travel time, cost of production). Any part of human activity is more or less optimized.

In real applications we focus on the methods of searching the minimum of function, because myltiplying the function by -1 transforms maximum into minimum.

For further purpose we assume the following definition of **global minimum of function**

$$f : \mathbb{R}^n \rightarrow \mathbb{R}$$

$$\min \ f(\vec{x}) = f(\vec{x}^*) \Leftrightarrow \bigwedge_{\vec{x} \in R^n} f(\vec{x}^*) \leq f(\vec{x})$$

$$\vec{x} = [x_1, x_2, \ldots, x_n]^T$$

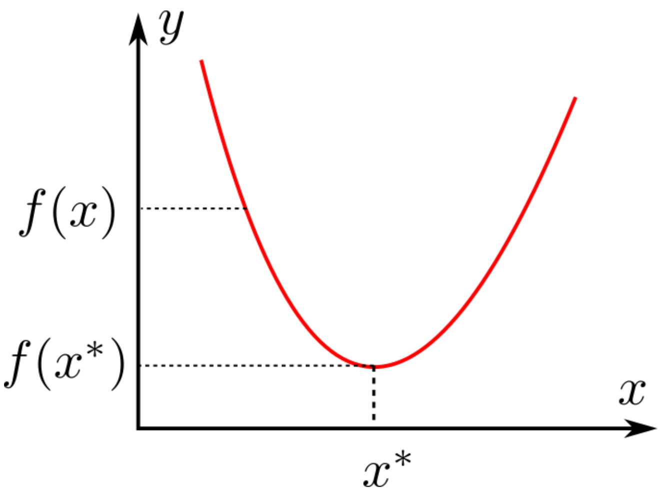
The function f is called an objective or target function

Considered minimization problem will become more complex if we impose some constraints on the solution, these are usually defined by set of scalar functions

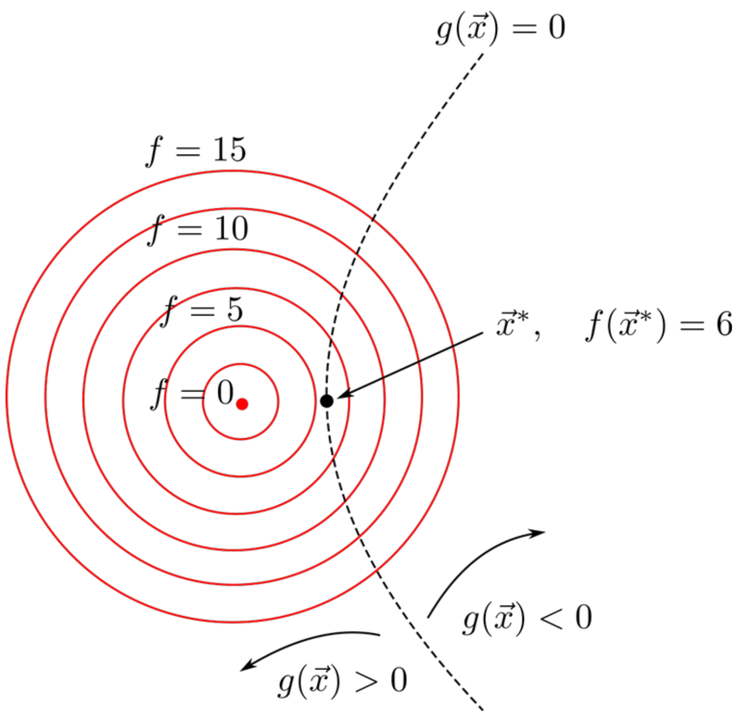
$$g_j : \mathbb{R}^n \rightarrow \mathbb{R}, \quad g_j(\vec{x}) \leq 0, \quad j = 1, 2, \ldots, m$$

$$h_j : \mathbb{R}^n \rightarrow \mathbb{R}, \quad h_j(\vec{x}) = 0, \quad j = 1, 2, \ldots, r$$

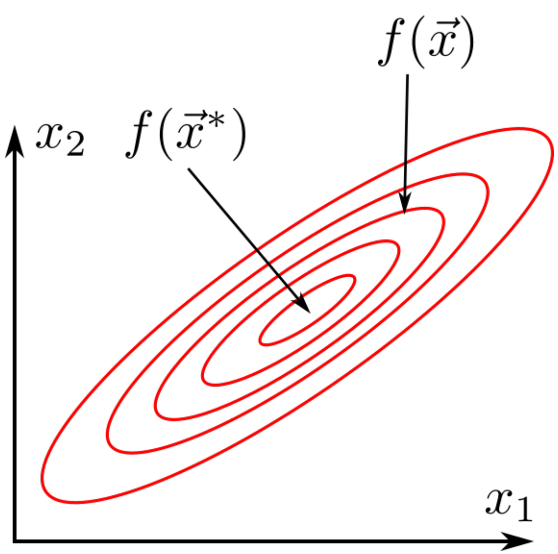
- **1D problem**



- **2D problem with constraints**



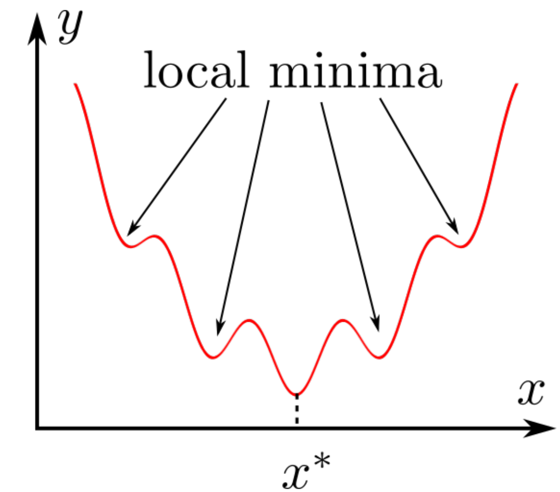
- **2D problem**



Global and local minima, saddle point

Besides the **global minimum**

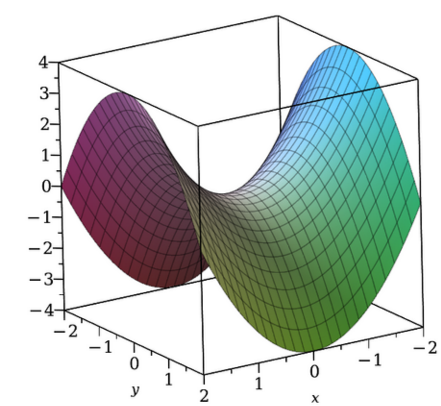
$$\bigwedge_{\vec{x}\in R^n} f(\vec{x}) \geq f(\vec{x}^*)$$



we define also **local minimum**

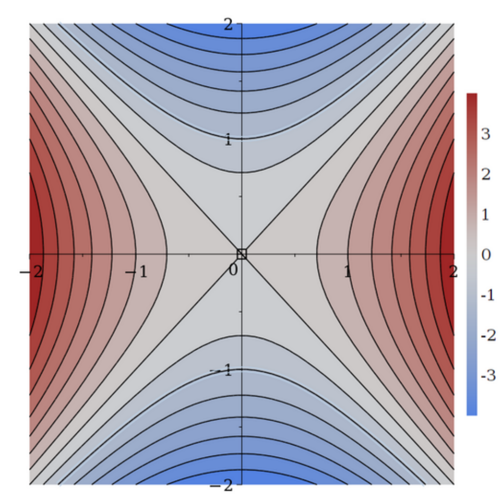
$$\exists\ \varepsilon : \varepsilon > 0, \varepsilon \in \mathbb{R} \quad \bigwedge_{\vec{x}:\|\vec{x}-\vec{x}^{\,local}\|<\varepsilon} f(\vec{x}) > f(\vec{x}^{\,local})$$

and the **saddle point**



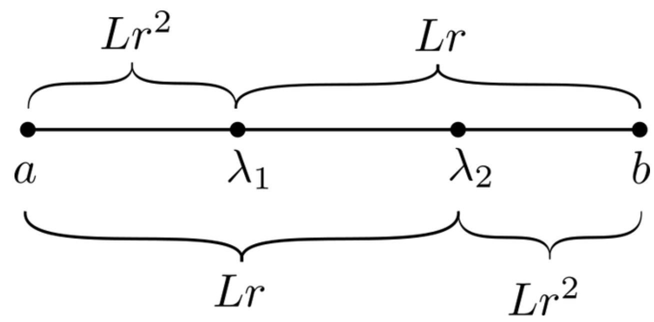
$$\vec{x}^{\,saddle} = \begin{pmatrix} \vec{x}^{\,0} \\ \vec{y}^{\,0} \end{pmatrix}$$

$$\exists\ \varepsilon : \varepsilon > 0, \varepsilon \in \mathbb{R} \quad \bigwedge_{\substack{\vec{x}:\|\vec{x}-\vec{x}^{\,0}\|<\varepsilon \\ \vec{y}:\|\vec{y}-\vec{y}^{\,0}\|<\varepsilon}} f(\vec{x}, \vec{y}^{\,0}) \leq f(\vec{x}^{\,0}, \vec{y}^{\,0}) \leq f(\vec{x}^{\,0}, \vec{y})$$



Golden Ratio

Ancient Greeks said that the segment can be divided into three parts with kept ratio between longer and shorter segments as follows



(*)

$$\frac{(\lambda_1 - a) + (b - \lambda_1)}{b - \lambda_1} = \frac{b - \lambda_1}{\lambda_1 - a} = \varphi$$

What are λ_1 and λ_2?

(1)

$$b - a = L \quad \Rightarrow \quad b = L + a$$

$$\frac{L}{L + a - \lambda_1} = \frac{L + a - \lambda_1}{\lambda_1 - a}$$

$$L(\lambda_1 - a) = (L - (\lambda_1 - a))^2$$

$$(\lambda_1 - a) = L \underbrace{\left(1 - \frac{(\lambda_1 - a)}{L}\right)^2}_{=r^2} = Lr^2$$

(2)

$$b - \lambda_1 = L - (\lambda_1 - a)$$

$$b - \lambda_1 = L\left(1 - \frac{(\lambda_1 - a)}{L}\right) = Lr$$

$$\frac{Lr^2 + Lr}{Lr} = \frac{Lr}{Lr^2} = \frac{1}{r} \quad \Rightarrow \quad r^2 + r - 1 = 0$$

$$r = \frac{\pm\sqrt{5} - 1}{2}$$

$$r_+ = \frac{\sqrt{5} - 1}{2} = 0.618034 > 0$$

$$\lambda_1 = a + r^2 L \qquad \lambda_2 = a + rL$$

Golden Section method

It is iterative method for univariate problems, in each iteration we narrow the area that contains the minimum until the width drops below some threeshold which determines the uncertainty of localization the point in question.

This procedure works as follows

- initialization: define the region of seeking the minimum

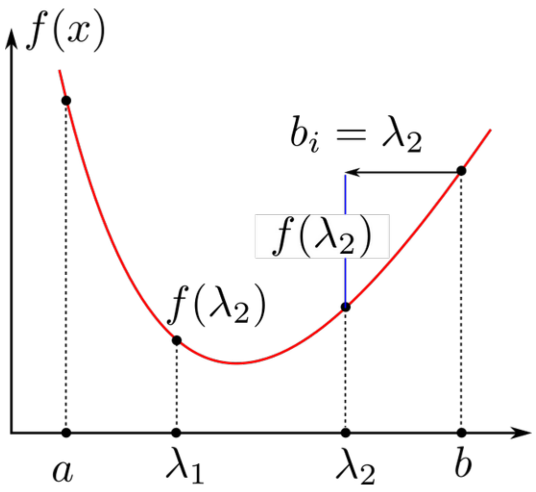
$$x \in [a_0, b_0]$$

- golden ratio division: in the segment [a_i,b_i] choose two points λ_1 and λ_2, compare values of objective function at these points

$$f(\lambda_2) > f(\lambda_1) \quad \vee \quad f(\lambda_2) < f(\lambda_1)$$

- narrowing the segment: reject the point of lower value and replace the nearest endpoint a_i or b_i with the point of larger value of objective function

$$\lambda_1 \to a_i \quad \vee \quad \lambda_2 \to b_i$$



- reccurence: repeat the last two steps until the width of searching interval falls below threshold ε

$$|a_i - b_i| < \varepsilon$$

then stop the process and assume the minimum is in center of this interval

$$x^* = \frac{a_i + b_i}{2}$$

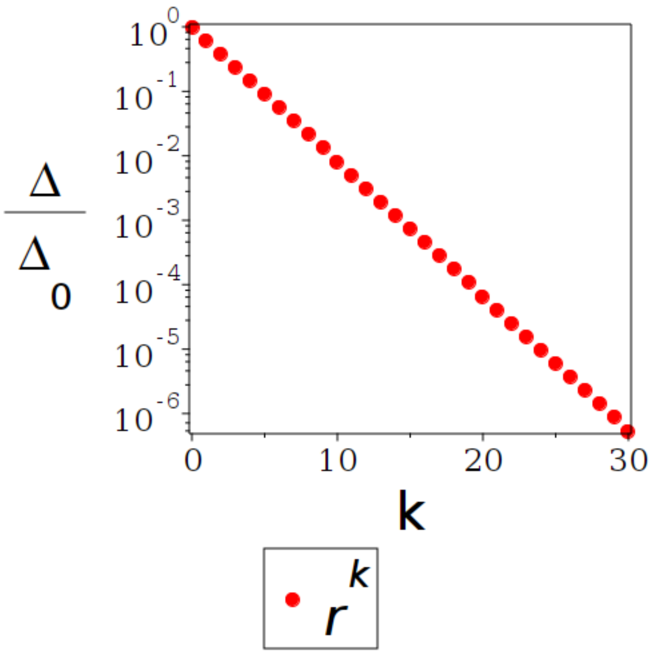
- pseudocode of Golden Section method

input: a, b, ε, $r = 0.618$, itmax, k=0
do
    k++
    $L \leftarrow b - a$
    $\lambda_1 \leftarrow r^2 L$
    $\lambda_2 \leftarrow rL$
    if $f(a+\lambda_1) > f(a+\lambda_2)$ then
        $a \leftarrow \lambda_1$
    else
        $b \leftarrow \lambda_2$
    end if
while $|a-b| > \varepsilon$ and $k < itmax$

$x^* \leftarrow \frac{a+b}{2}$

- efficency of the method

$$\frac{\Delta_k}{\Delta_0} = \frac{|b_k - a_k|}{|b_0 - a_0|} = r^k$$



Powell square-interpolation method

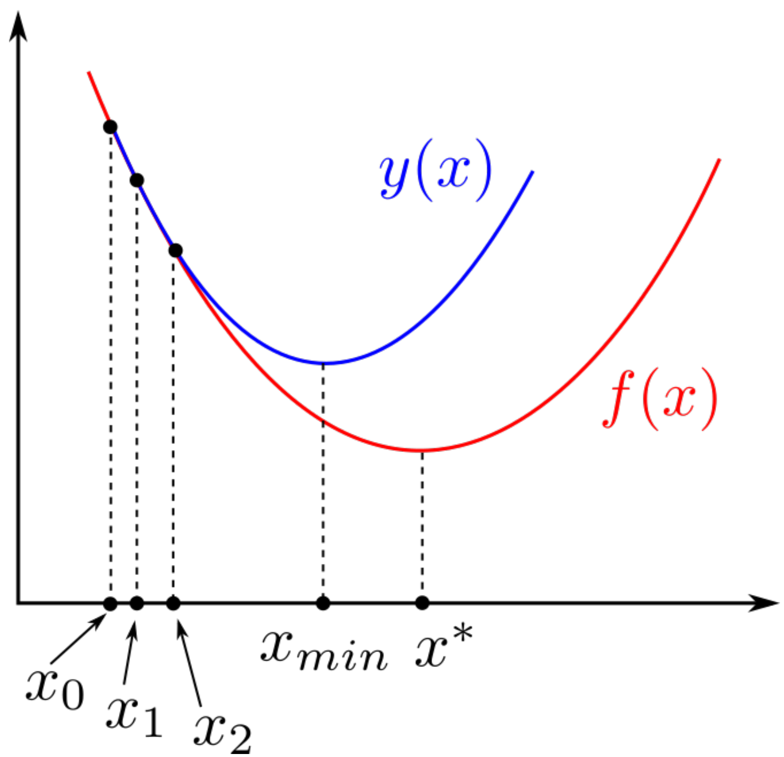
We assume that the objective function can locally be interpolated well with parabolic function (there are needed three points: x_0, x_1, x_2)

$$\{(x_0, f(x_0)),\ (x_1, f(x_1)),\ (x_2, f(x_2))\}$$

$$y(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1)$$

Coefficients of polynomial are expressed by means of finite differences

$$\boxed{\begin{aligned} a_0 &= f(x_0) \\ a_1 &= \frac{f(x_1) - f(x_0)}{x_1 - x_0} \\ a_2 &= \frac{\frac{f(x_2)-f(x_1)}{x_2-x_1} - \frac{f(x_1)-f(x_0)}{x_1-x_0}}{x_2 - x_0} \end{aligned}}$$



We treat the minimum of this parabola as approximate position of object function's minimum

$$\frac{dy}{dx} = a_1 + 2xa_2 - a_2(x_0 + x_1) = 0$$

$$\boxed{x_{\min} = \frac{a_2(x_0 + x_1) - a_1}{2a_2} \approx x^*}$$

Single iteration of Powell interpolation is a few step procedure

- for given three interpolation points, sorted in ascending order find the minimum of parabola

$$x_0 < x_1 < x_2 \quad \Longrightarrow \quad x_{\min}$$

- among the three points x_0, x_1, x_2 find the more distant from the minimum x_{min} and reject it, it shall be x_0 or x_2
- attach x_{min} to the rest two points and sort them in ascending order
- check if the distance between last two approximations of objective function's minima would be accpeted or not

```
input:  x⃗ = [x0, x1, x2] ,  ε ,  itmax ,  k=0,  x_old = large number
do
     k++
     !calculate:
     a1 ,  a2 ,  x_min
     !find and reject the most distant point from x_min
     if |x0 − x_min| > |x2 − x_min| then
          x0 = x_min
     else
          x2 = x_min
     end if
     sort(x⃗, ascending)
     !check convergence
     s = |x_min − x_old|
     x_old = x_min
while s > ε and k < itmax
```

Gradient methods

Let the vectors **x** and **u** be fixed a present approximation and a new direction of searching in next iteration, then

$$F:\mathbb{R}\to\mathbb{R},\quad \lambda\in\mathbb{R} \quad \to \quad f(\vec{x}_{i+1}) = f(\vec{x}_i+\lambda\vec{u}) \equiv F(\lambda)$$

$$f(\vec{x}_i) = F(0)$$

to find out the value of F(λ) we use Taylor series

$$F(0+\lambda) = F(0) + \frac{dF}{d\lambda}\lambda + O(\lambda^2)$$

We assume **λ>0** and looking for the condition which makes the first derivative of F to be negative.

$$\begin{aligned} f(\vec{x}_{i+1}) &\equiv F(\lambda) \\ df &\equiv dF \\ df &= \nabla^T f(\vec{x}_i) d\vec{x} = \nabla^T f(\vec{x}_i)\, \vec{u} d\lambda = dF \\ \frac{dF}{d\lambda} &= \nabla^T f(\vec{x}_i)\, \vec{u} \end{aligned}$$

$$F(0+\lambda) = F(0) + \underbrace{\lambda}_{>0} \underbrace{\nabla^T f\, \vec{u}}_{<0} = F(0) - \lambda\, |\nabla^T f\, \vec{u}| < F(0) = f(\vec{x}_i)$$

- we've showed that the new position should decrease the value of objective function
- gradient of f is fixed so how to choose direction **u** to meet this condition?

the simplest (not optimal)choice: $\vec{u} = -\nabla f$

Gradient methods change iteratively the initial solution

$$\vec{x}_0 \to \vec{x}_1 \to \vec{x}_2 \to \ldots \to \vec{x}_n$$

until one of convergence condition is met

$$\|\vec{x}_{i+1} - \vec{x}_i\|_2 < \varepsilon$$

common choice, simple interpretation in context of distance measure in euclidean space

$$\|\nabla f(\vec{x}_i)\|_2 < \varepsilon$$

$$|f(\vec{x}^{\,i}) - f(\vec{x}^{\,i-1})| < \varepsilon$$

less common, requires the knowledge of the gradient of function values scale, not reliable for flat minima

or the maximal number of iteration were reached indicating lack of convergence.

Remarks:

- iterative methods do not guarantee we will success when looking for the global minimum of the objective function, the larger is the dimensionality of the problem in question, more likely we get to the local minimum
- if we are not satisfied with present solution (lack of convergence or local minimum) we may start again the algorithm with initial vector chosen differently

Steepest descent method

It is the simplest gradient method, as a direction of searching for next iteration we choose the gradient of objective function

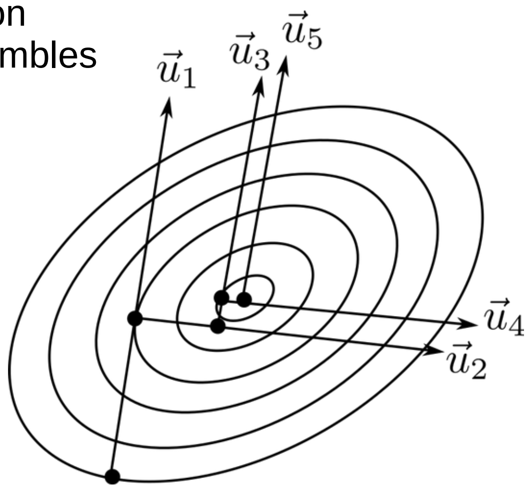
$$\vec{x}_{i+1} = \vec{x}_i + \lambda \vec{u} \qquad\qquad \vec{u} = \frac{-\nabla f(\vec{x}_i)}{\|\nabla f(\vec{x}_i)\|}$$

Since the direction **u** is fixed we must choose value of λ

- It can be set cautiously as a small number but it slows down an algorithm
- we may apply one of the one-dimensional methods, i.e. Golden Ratio or Powell interpolation, to function F(λ)

$$\boxed{F(\lambda) = f(\vec{x}_i + \lambda \vec{u}) = \min_{\lambda} f(\vec{x}_i + \lambda \vec{u})}$$

One may expect the latter would be very effective, however, directional optimization makes that the consequtive directions are orthogonal and the searching path resembles a zig-zag trajectory rather than an expected optimal smooth one.



$$\frac{dF(\lambda)}{d\lambda} = 0 \quad \Longrightarrow \quad \frac{\partial f(\vec{x}_i + \lambda \vec{u}_i)}{\partial \lambda} = \left(\nabla f(\vec{x}_i + \lambda \vec{u}_i)\right)^T \vec{u}_i = 0$$

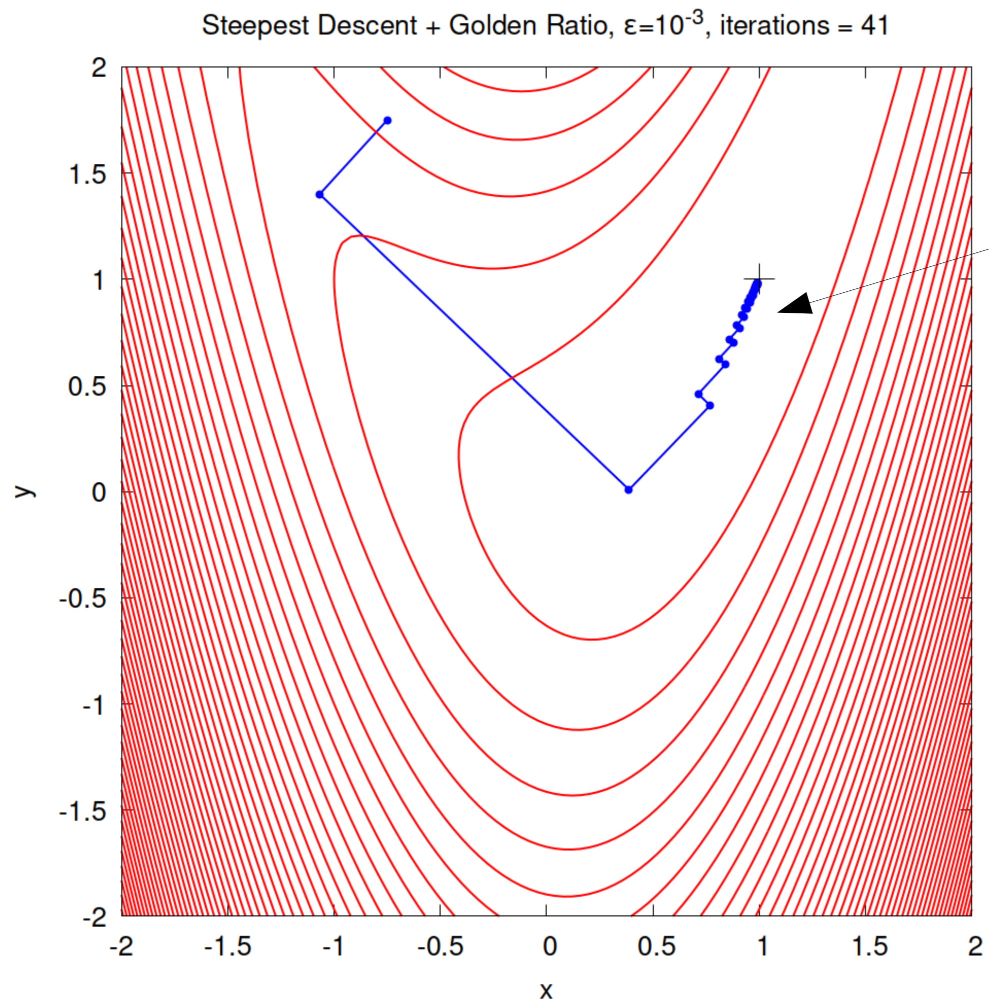
$$\nabla f(\vec{x}_i + \lambda \vec{u}_i) = \nabla f(\vec{x}_{i+1}) = -\vec{u}_{i+1}$$

$$\boxed{\vec{u}_{i+1}^T \cdot \vec{u}_i = 0}$$

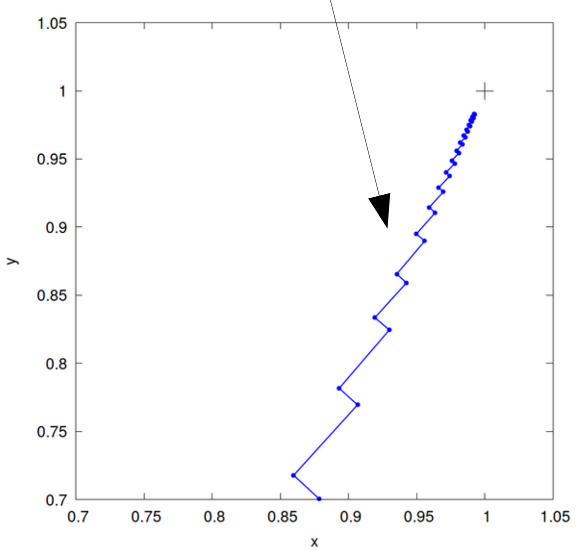
Example: Steepest Descent method in conjunction with Golden Ratio in searching the minimum of the objective function

$$f(x,y) = \frac{5}{2}(x^2 - y)^2 + (1-x)^2$$

$$\nabla f = \begin{pmatrix} 10(x^2 - y)x + 2x - 2 \\ -5x^2 + 5y \end{pmatrix}$$



zigzag occurs hindering the motion towards the minimum



Congugate Gradients method

The basic version of CG method for solving SLE has been worked out for the quadratic problem i.e. the function f contains at most squares of elements of solution vector

$$f : \mathbb{R}^n \to \mathbb{R}, \quad A \in \mathbb{R}^{n\times n}, \quad \vec{x}, \vec{b} \in \mathbb{R}^n, \quad c \in \mathbb{R}$$

$$f(\vec{x}) = \frac{1}{2}\vec{x}^T A\vec{x} + \vec{x}^T\vec{b} + c > 0$$

However, when solving the minimization problem in most cases we operate on functions which can be only approximated locally by quadratic form. We don't even know the form of the matrix A (Hessian), however we may calculate the downhill gradient directly differentating function f(x).

The iterative CG procedure of minimization of objective function f(x) is following

$$\begin{aligned}
\vec{g}_i &= -\nabla f(x_i) \\
\gamma_{i-1} &= \frac{\vec{g}_i^T \vec{g}_i}{\vec{g}_{i-1}^T \vec{g}_{i-1}} \quad \text{(Fletcher-Reeves)} \\
\gamma_{i-1} &= \frac{(\vec{g}_i - \vec{g}_{i-1})^T \vec{g}_i}{\vec{g}_{i-1}^T \vec{g}_{i-1}} \quad \text{(Polak-Ribiere)} \\
i = 0: \quad \vec{u}_i &= \vec{g}_i \\
i > 0: \quad \vec{u}_i &= \vec{g}_i + \gamma_{i-1}\vec{u}_{i-1} \\
\lambda &\leftarrow \min f(x_i + \lambda\vec{u}_i), \quad \text{(Golden Section)} \\
\vec{x}_{i+1} &= \vec{x}_i + \lambda\vec{u}_i,
\end{aligned}$$

Conjugate Gradients method

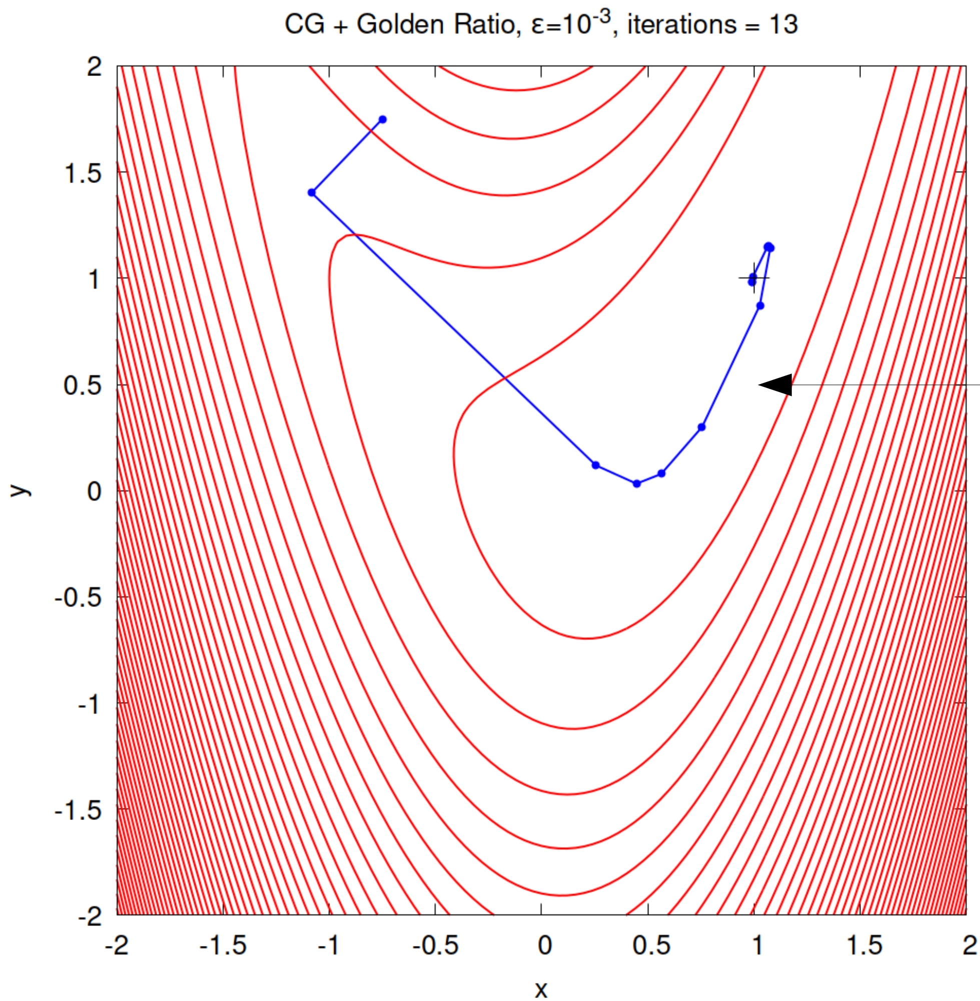
```
input:  x⃗ = [x_0, x_1, ..., x_n], ε, itmax, k=0
do
    g⃗ ← −∇ f(x⃗)
    if k=0 then
        u⃗ ← g⃗
    else
        γ ← (g⃗^T g⃗) / (g⃗_old^T g⃗_old)
        u⃗ ← g⃗ + γ u⃗
    end if
!save g⃗ for next iteration
    g⃗_old ← g⃗
    k++
!perform univariate minimization
    λ ← min f(x⃗ + λu⃗)  (Fletecher−Reeves)
!save x⃗_old for checking the convergence
    x⃗_old ← x⃗
!update solution
    x⃗ ← x⃗ + λu⃗
    s ← |x⃗ − x⃗_old|
while s > ε and k < itmax
```

Example: application of the CG method in searching the minimum of objective function

$$f(x,y) = \frac{5}{2}(x^2 - y)^2 + (1-x)^2$$

it is bivariate 4-th order polynomial

$$\nabla f = \begin{pmatrix} 10(x^2 - y)x + 2x - 2 \\ -5x^2 + 5y \end{pmatrix}$$



searching path changes trajectory in less abruptly than SD method, less iterations are needed to succeed

Newton Method

Let's consider standard iterative formula used in gradient-based minimization methods

$$\vec{x}_{i+1} = \vec{x}_i + \vec{u}$$

here we skip scaling factor λ,
we add it heuristically at the end

At the minimum of objective function its gradient vanishe. Assuming it happens in next iteration

$$\nabla f(\vec{x}_{i+1}) = \vec{0}$$

we may find the link between these two iterations from the analysis of the local changes near the minimum.
For this purpose we need to expand gradient into Taylor series

$$\nabla f(x_{i+1}) = \nabla f(\vec{x}_i) + \underbrace{\left(\nabla \otimes \nabla^T f(x_i)\right)}_{H(\vec{x})} \vec{u} + O\left(\|\vec{u}\|^2\right)$$

$$\nabla f(\vec{x}_i) = -H(\vec{x})\vec{u} \qquad (-H^{-1}) \cdot /$$

$$\vec{u} = -H(\vec{x})\nabla f(\vec{x}_i)$$

$$\boxed{\vec{x}_{i+1} = \vec{x}_i - \lambda H^{-1}(\vec{x})\nabla f(\vec{x}_i)}$$

We get an iterative formula with ad hoc added scaling factor λ
which value shall be optimized by one of univariate
minimization methods.

$$H \in \mathbb{R}^{n \times n} \quad \text{- Hessian}$$

$$H(\vec{x}) = \nabla \otimes \nabla^T f(\vec{x})$$

$$= \begin{bmatrix} \frac{\partial^2 f(\vec{x})}{\partial x_1^2} & \frac{\partial^2 f(\vec{x})}{\partial x_1 \partial x_2} & \cdots & \cdots \\ \frac{\partial^2 f(\vec{x})}{\partial x_2 \partial x_1} & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 f(\vec{x})}{\partial x_n \partial x_1} & \cdots & \cdots & \frac{\partial^2 f(\vec{x})}{\partial x_n^2} \end{bmatrix}$$

Newton method

```
input:  x⃗ = [x_0, x_1, ..., x_n] , ε , itmax , k=0
do
    k++
    g⃗ ← ∇f(x⃗)
    for i=0 to n−1 by 1 do
        for j=i to n−1 by 1 do
        H_{i,j} = ∂²f(x⃗)/(∂x_i ∂x_j)
!symmetry of mixed derivatives
        H_{j,i} = H_{i,j}
        end do
    end do
!save x⃗_old for checking the convergence
    x⃗_old ← x⃗
    H^{-1} ← inverse(H)
    u⃗ = −H^{-1} g⃗
!perform univariate minimization
    λ ← min f(x⃗ + λu⃗)
!update solution
    x⃗ ← x⃗ + λu⃗
    s ← |x⃗ − x⃗_old|
while s > ε and k < itmax
```

Example: use of Newton method in searching the minimum of objective function

$$f(x,y) = \frac{5}{2}(x^2 - y)^2 + (1-x)^2$$

- gradient

$$\nabla f = \begin{pmatrix} 10(x^2 - y)x + 2x - 2 \\ -5x^2 + 5y \end{pmatrix}$$

- Hessian

$$H(x,y) = \begin{pmatrix} 30x^2 - 10y + 2 & -10x \\ -10x & 5 \end{pmatrix}$$

Results for three methods:

Steepest Descent – **41** iterations
Conjugate Gradients – **13** iteration
Newton – **8** iterations