L. Function minimization and optimization

V. Blobel

Inst. für Experimentalphysik, Hamburg

Sommersemester 2002

1. Optimization

Optimization problems: set of independent variables (parameters) $\boldsymbol{x}$ with n components, and, sometimes, in addition constraints $c_i(\boldsymbol{x})$ (conditions or restrictions) that define acceptable (allowed) values of the variables.

Essential component of an optimization problem:

a single measure of "goodness", termed objective function $F(\boldsymbol{x})$

which depends in some way on the variables $\boldsymbol{x}$.

In mathematical terms: optimization involves *maximizing* or *minimizing*, for example "profit" or "weight".

In mathematical terms:

$$\begin{array}{rlll} \text{minimize} & F(\boldsymbol{x}) & \boldsymbol{x} \in \mathcal{R}^n & \\ \text{subject to} & c_i(\boldsymbol{x}) = 0, & i = 1, 2, \ldots m' \\ & c_i(\boldsymbol{x}) \geq 0, & i = m' + 1, \ldots m \end{array}$$

Optimization in Physics data analysis

Objective function $F(\boldsymbol{x})$ is

- (negative log of a) Likelihood function in the Maximum Likelihood method, or
- sum of squares in a (nonlinear) Least Squares problem

Constraints $c_i(\boldsymbol{x})$ are

- equality contraints, expressing relations between parameters, and
- inequality constraints are limits of certain parameters, defining a restricted range of parameter values (e.g. $m_\nu^2 \geq 0$).

Sometimes there are two objective functions: $F(\boldsymbol{x})$, requiring a good data description, and $G(\boldsymbol{x})$, requiring e.g. smoothness:

$$\text{minimize sum} \quad F(\boldsymbol{x}) + \tau \cdot G(\boldsymbol{x})$$

with *weight* parameter τ.

Aim of optimization

- Find the global minimum of the objective function $F(\boldsymbol{x})$,
- within the allowed range of parameter values,
- in a short time even for a large number of parameters and a complicated objective function,
- even is there are local minima.

Most methods will converge to the *next* minimum, which may be the global minimum or a local minimum (corresponding to *rapid* cooling), going immediately downhill as far as possible.

Search for **the** global minimum requires a special effort.

A special case: combinatorical minimization

The travelling salesman problem:

> Find the shortest cyclical itenary for a travelling salesman who must visit each of N cities in turn.

Design of complex integrated circuits:

> Find the optimal arrangement of a large number of circuit elements to minimize interference among their connecting wires.

Track finding from hits, vertex finding from tracks in HEP

Objective function is e.g. the total path length.

Space over which the function is defined is a discrete, but very large, configuration space (e.g. set of possible orders of cities), with a number of elements in the configuration space which is factorial large (remember: $69! \approx 10^{100}$). The concept of a "direction" may have no meaning.

combinatorical minimization, contnd.

Desired global extremum may be hidden among many, poorer, local extrema.

> Nature is, for slowly cooled systems, able to find the minimum energy state of a crystal that is completely ordered over a large distance.

The method: Simulated annealing, in analogy with the way metals cool and anneal, with slow cooling, using Boltzmanns probability distribution

$$\text{Prob}(E) \sim \exp\left(-E/k_B T\right)$$

for a systen in equilibrium.

Even at low T there is a chance for a high energy state, with a corresponding chance to get out of a local minimum.

The Metropolis algorithm: the elements

- A description of possible system configurations.
- A generator of random changes in the configuration (called “option”).
- An objective function E (analog of energy) whose minimizatiuon is the goal of the procedure.
- A control parameter T (analog of temperature) and an *annealing* schedule (for cooling down).

A random option will change the configuration from energy E_1 to energy E_2. This option is accepted with probability

$$p = \exp\left[\frac{-(E_2 - E_1)}{k_b T}\right]$$

Note: if $E_2 < E_1$ the option is always accepted.

Example: the “effective” solution of the travelling salesman problem

2. One-dimensional minimization

Search for minimum of function $f(x)$ of (scalar) argument x.

Important application in multidimensional minimization: robust minimization *along a line*

$$f(z) = F(\boldsymbol{x} + z \cdot \boldsymbol{\Delta x}) \qquad \text{Line search}$$

Aim: robust, efficient and as fast as possible, because each function evaluation my require a large cpu time.

Standard method: iterations, starting from x_0, with expression $\Phi(x)$:

$$x_{k+1} = \Phi(x_k) \quad \text{for} \quad k = 0, 1, 2, \ldots$$

with convergence to fixed point x^* with $x^* = \Phi(x^*)$

Newtons iteration method, based on derivatives

Method for the determination of zeros of a function ($f(x) = 0$):

$$x_{k+1} = \Phi(x_k) \quad \text{with} \quad \Phi(x) = x - \frac{f(x)}{f'(x)} \quad \text{for} \quad k = 0, 1, 2, \ldots$$

Same method for min/max determination ($f'(x) = 0$):

$$x_{k+1} = \Phi(x_k) \quad \text{with} \quad \Phi(x) = x - \frac{f'(x)}{f''(x)} \quad \text{for} \quad k = 0, 1, 2, \ldots ,$$

derived from Taylor expansion

$$f'(x+h) \approx f'(x) + h\, f''(x)$$

It follows:

$$\boxed{x_{k+1} = \Phi(x_k) = x_k + h \qquad \text{with} \qquad h = - \left.\frac{f'(x)}{f''(x)}\right|_{x=x_k}}$$

Convergence behaviour of iterative methods

An iterative method $x_{k+1} = \Phi(x_k)$ is called *local convergent* of at least order p, if for all start values $x_0 \in U(x^*)$

$$|x_{k+1} - x^*| < c \cdot |x_k - x^*|^p$$

is valid for all k ($c < 1$ in the linear case p =1).

Condition for order p:

$$\begin{array}{rrcll} \text{If} & \Phi^{(j)}(x^*) & = & 0 & \text{for all } j = 1, \ldots (p-1) \\ \text{and} & \Phi^{(p)}(x^*) & \neq & 0 & \end{array}$$

$$\text{then} \quad \lim_{k \to \infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^p} = \frac{\Phi^{(p)}(x^*)}{p!}$$

and the iterative method is convergent of order p.

Convergence behaviour contnd.

The linear case ($p = 1$): $|\Phi'(x^*)| < 1$ required.

Sequence x_k converging monotonely to x^* for a positive value of $\Phi'(x^*)$, and alternating around x^* for a negative value.

linear convergence can be very slow.
The constant c is often very close to 1, and many 100 iterations may be necesary with small progress per iteration
– not recommended.

quadratic convergence usually only few iterations required, very fast in final phase – recommended, at least for the *end game*.

Convergence behaviour of Newtons method ...

$$\Phi(x) = x - \frac{f'(x)}{f''(x)}$$

...for the determination of a minimum (or maximum):

$$\Phi'(x^*) = 1 - \frac{(f''(x^*))^2 - f'(x^*)\, f'''(x^*)}{(f''(x^*))^2} = \frac{f'(x^*)\, f'''(x^*)}{(f''(x^*))^2} = 0$$

$$\Phi''(x^*) = \frac{f'''(x^*)}{f''(x^*)} \neq 0 \quad \text{in general}$$

Newtons method:

- quadratically convergent (locally);
- first and second derivative required;
- may be divergent for a bad start value.

Alternative method: search without derivatives

Required: robust convergent method for minimum determination without the need to calculate derivatives (which may be complicated or impossible).

Aim: determine very short x-interval, which contains the minimum of the function $f(x)$.

Strategy of search method with two steps:

(1) find initial interval: Find interval $[x_a, x_b]$, which includes minimum: $x_a < x_{\text{inner}} < x_b$
Asssume function to be unimodal: minimum included, if $f(x_{\text{inner}}) < f(x_a)$ and $f(x_{\text{inner}}) < f(x_b)$ for some x_{inner}.

(2) reduce size of interval: reduce the size of the interval sufficiently.

A simple search method

(1) Find initial interval: Start with initial values x_1, x_2 with $f(x_1) > f(x_2)$ and evaluate function value at

$$x_i = x_{i-1} + \alpha(x_{i-1} - x_{i-2}) \qquad i = 3, 4, \ldots \qquad \alpha > 1 \,,$$

until a function value

$$f(x_i) > f(x_{i-1})$$

is found. Then a minimum is included in interval $[x_{i-2}, x_i]$.

Reduction of the interval: The interval $[x_{i-2}, x_i]$ is divided by an inner point x_{i-1} with

$$f(x_{i-1}) < f(x_{i-2}) \qquad\qquad f(x_{i-1}) < f(x_i)$$

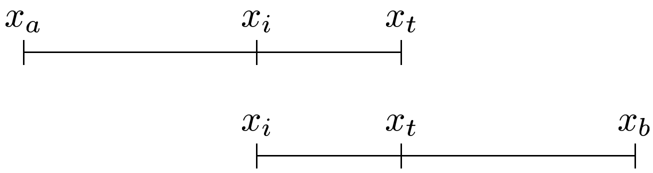
Based on one additional function value at another inner point x_t, the interval can be reduced.

Golden section strategy for interval reduction

$$\frac{x_b - x_i}{x_b - x_a} = \frac{x_i - x_a}{x_b - x_i} = \tau \qquad \to \tau = \left(\sqrt{5} - 1\right)/2 = 0.618034$$

Define new point x_t by $(x_t - x_a)/(x_b - x_a) = \tau$:

New interval, depending on function value $f(x_t)$:



Reduction of length of interval by factor $\tau = 0.618034$ by one function value ($\to$ linear convergence).
For 10 function evaluations reduction by factor $\tau^{10} \approx 0.01$, not dependent on function behaviour.

Alternative to golden section

More efficient for “normal” behaviour of functions: fit parabola to last three function values and use minimum of parabola as next point.

Note: Many functions to be minimized are parabolic in good approximation $\to$ minimum of parabola often close to function minimum.

But: method can get stuck with unbalanced section of the interval;

$\to$ Combined method: use mixture of parabola and golden section method, to avoid unbalanced section of the interval.

3. Search methods in n dimensions

Search methods in n dimensions do not require any derivatives, only functions values.

Examples:

Line search in one variable, sequentially in all dimensions
- this is usually rather inefficient

Simplex method by Nelder and Mead: simple, but making use of earlier function evaluations in an efficient way ("learning")

Monte Carlo search: random search in n dimensions, using result as starting values for more efficient methods
– meaningful if several local minima may exist

In general search methods are acceptable initially (far from the optimum), but are inefficient and slow in the "end game".

Simplex method in n dimensions

A simplex is formed by $n+1$ points in n-dimensional space ($n=2$ triangle)

$$\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \ldots \boldsymbol{x}_n,\ \boldsymbol{x}_{n+1}$$

sorted such that function values $F_j = F(\boldsymbol{x}_j)$ are in the order

$$F_1 \leq F_2 \leq \ldots F_n \leq F_{n+1}$$

In addition: mean of best n points =

$$\text{center of gravity} \qquad \boldsymbol{c} = \frac{1}{n}\sum_{j=1}^{n} \boldsymbol{x}_j$$

Method: sequence of cycles with new point in each cycle, replacing worst point x_{n+1}, with new (updated) simplex in each cycle.

At the start of each cycle new test point $\boldsymbol{x}_r$, obtained by reflexion of worst point at the center of gravity:

$$\boldsymbol{x}_r = \boldsymbol{c} + \alpha \cdot (\boldsymbol{c} - \boldsymbol{x}_{n+1})$$

A cycle in the Nelder and Mead method

Depending on value $F(x_r)$:

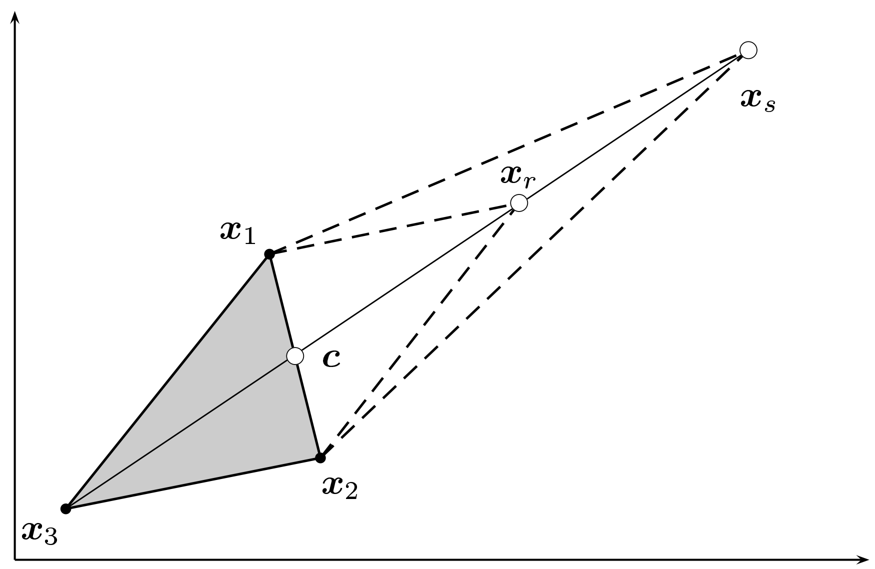
[$F_1 \leq F_r \leq F_n$]: Test point $\boldsymbol{x}_r$ is middle point and is added, the previous worst point is removed.

[$F_r \leq F_1$]: Test point $\boldsymbol{x}_r$ is best point, search direction seems to be effective. A new point $\boldsymbol{x}_s = \boldsymbol{c} + \beta \cdot (\boldsymbol{x}_r - \boldsymbol{c})$ (with $\beta > 1$) is determined and the function value is F_s evaluated. For $F_s < F_r$ the extra step is successful, $\boldsymbol{x}_{n+1}$ is replaced by $\boldsymbol{x}_s$ otherwise by $\boldsymbol{x}_r$.

[$F_r > F_n$]: the simplex is too big and has to be reduced in size. For $F_r < F_{n+1}$ the test point $\boldsymbol{x}_r$ replaces the words point $\boldsymbol{x}_{n+1}$. A new test point $\boldsymbol{x}_s$ is defined by $\boldsymbol{x}_s = \boldsymbol{c} - \gamma(\boldsymbol{c} - \boldsymbol{x}_{n+1})$ (contraction) with $0 < \gamma < 1$. If this point $\boldsymbol{x}_s$ with $F_s = F(\boldsymbol{x}_s) < F_{n+1}$ is an improvement, then $\boldsymbol{x}_{n+1}$ is replaced by this point $\boldsymbol{x}_s$. Otherwise a new simplex in defined by replacing all points except $\boldsymbol{x}_1$ by $\boldsymbol{x}_j = \boldsymbol{x}_1 + \delta(\boldsymbol{x}_j - \boldsymbol{x}_1)$ for $j = 2, \ldots n+1$ with $0 < \delta < 1$, which requires n function evaluations.

Typical values are $\alpha = 1$, $\beta = 2$, $\gamma = 0.5$ und $\delta = 0.5$.

The Simplex method



A few steps of the Simplex method, starting from the simplex $(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)$ with center of gravity $\boldsymbol{c}$. The points $\boldsymbol{x}_r$ and $\boldsymbol{x}_s$ are test points.

Monte Carlo search in n dimensions

Search in a box. Lower and upper boundaries $x_i^{(l)}$ and $x_i^{(u)}$ defined for each dimension

$$\text{test point} \qquad x_i^{(t)} = r_i\, x_i^{(l)} + (1 - r_i)\, x_i^{(u)} \qquad i = 1, 2, \ldots n$$

with $r_i \in U(0,1)$ (uniformly distributed).

Search in a sphere. Direction cosinus d_i of search vector with $d_i \cdot \Delta x_i$

$$d_i = \frac{r_i}{(r_1^2 + r_2^2 + \ldots r_n^2)^{1/2}} \qquad i = 1, 2, \ldots n$$

with $r_i \in N(0,1)$ (from standard normal distribution).

Meaningful in higher dimensions, especially if existence of many local minima expected, as method to get good starting values.

4. Unconstrained minimization

$$\boxed{\text{minimize } F(\boldsymbol{x}) \qquad \boldsymbol{x} \in \mathcal{R}^n}$$

Taylor expansion:

$$\begin{array}{lrcl} \text{function} & F(\boldsymbol{x}_k + \boldsymbol{\Delta x}) & \approx & F_k + \boldsymbol{g}_k^T \boldsymbol{\Delta x} + \frac{1}{2} \boldsymbol{\Delta x}^T \boldsymbol{H}_k \boldsymbol{\Delta x} \\ \text{derivative} & \boldsymbol{g}(\boldsymbol{x}_k + \Delta \boldsymbol{x}) & \approx & \boldsymbol{g}_k + \boldsymbol{H}_k \boldsymbol{\Delta x} \;, \end{array}$$

Function value and derivatives are evaluated at $\boldsymbol{x} = \boldsymbol{x}_k$:

$$\begin{array}{lrcll} \text{function} & F_k & = & F(\boldsymbol{x}_k) & \\ \text{gradient} & (g_k)_i & = & \left.\frac{\partial F}{\partial x_i}\right|_{\boldsymbol{x}=\boldsymbol{x}_k} & i = 1,\, 2,\, \ldots n \\ \text{Hesse matrix} & (H_k)_{ij} & = & \left.\frac{\partial^2 F}{\partial x_i \partial x_j}\right|_{\boldsymbol{x}=\boldsymbol{x}_k} & i,j = 1,\, 2,\, \ldots n \end{array}$$

Covariance matrix for statistical objective functions

Note: if the objective function is

- a sum of squares of deviations, defined according to the Method of Least Squares, or
- a negative log. Likelihood function, defined according to the Maximum Likelihood Method,

then the inverse Hessian $\boldsymbol{H}$ at the minimum is a good estimate of the covariance matrix of the parameters $\boldsymbol{x}$:

$$\boldsymbol{V}_{\boldsymbol{x}} \approx \boldsymbol{H}^{-1}$$

The Newton step

Step $\boldsymbol{\Delta x}_N$ determined from $\boldsymbol{g}_k + \boldsymbol{H}_k \boldsymbol{\Delta x} = 0$:

$$\boldsymbol{\Delta x}_N = -\boldsymbol{H}_k^{-1} \boldsymbol{g}_k \ .$$

For a quadratic function $F(\boldsymbol{x})$ the Newton step is, in length and direction, a step to the minimum of the function.

Sometimes large angle ($\approx 90°$) between Newton direction $\boldsymbol{\Delta x}_N$ and $\boldsymbol{g}_k$ (the direction of steepest descent).

Calculation of the distance ΔF to the minimum (called EDM in MINUIT):

$$d = -\boldsymbol{g}_k^T \cdot \Delta \boldsymbol{x}_N = \boldsymbol{g}_k^T \boldsymbol{H}_k^{-1} \boldsymbol{g}_k = \boldsymbol{\Delta x}_N^T \boldsymbol{H}_k \boldsymbol{\Delta x}_N > 0$$

if Hessian positive-definite. For a quadratic function the distance to the minimum is $d/2$.

General iteration scheme

Define initial value $\boldsymbol{x}_0$, compute $F(\boldsymbol{x}_0)$ and set $k = 0$

1. **Test for convergence:** If the conditions for convergence are satisfied, the algorithm terminates with $\boldsymbol{x}_k$ as the solution. The difference $F_{k-1} - F_k$ and d are used in the test.
2. **Compute a search vector:** A vector $\boldsymbol{\Delta x}$ is computed as the new search vector. The newton search vector is determined from $\boldsymbol{H}_k \boldsymbol{\Delta x}_N = -\boldsymbol{g}_k$.
3. **Line Search:** A one-dimensional minimization is done for the function $f(z) = F(\boldsymbol{x}_k + z \cdot \boldsymbol{\Delta x}_N)$. and z_{min} is determined. (this step is essential to get a stable method!)
4. **Update:** The point $\boldsymbol{x}_{k+1}$ is defined by $\boldsymbol{x}_k + z_{\text{min}} \cdot \boldsymbol{\Delta x}$, and k is increased by 1.

Descent direction $\quad \boldsymbol{g}_k^T \cdot \Delta \boldsymbol{x} < 0 \quad$ if Hessian positive-definite

Method of steepest descent

The search vector $\boldsymbol{\Delta x}$ is equal to the negative gradienten

$$\boldsymbol{\Delta x}_{\text{SD}} = -\boldsymbol{g}_k$$

- Step seems to be natural choice;
- only gradient required (no Hesse matrix) – good;
- no step size defined (in contrast to the Newton step) – bad;
- rate of convergence only linear:

$$\frac{F(\boldsymbol{x}_{k+1}) - F(\boldsymbol{x}^*)}{F(\boldsymbol{x}_k) - F(\boldsymbol{x}^*)} = c \approx \left(\frac{\lambda_{\text{max}} - \lambda_{\text{min}}}{\lambda_{\text{max}} + \lambda_{\text{min}}}\right)^2 = \left(\frac{\kappa - 1}{\kappa + 1}\right)^2 ,$$

(λ_{max} and λ_{min} are largest and smallest eigenvalue and κ condition number of Hesse matrix $\boldsymbol{H}$. For a large value of κ c close to one and slow convergence – very bad.

Optimal step size, if Hesse matrix known:

$$\boldsymbol{\Delta x}_{\text{SD}} = -\left(\frac{\boldsymbol{g}_k^T \boldsymbol{g}_k}{\boldsymbol{g}_k^T \boldsymbol{H}_k \boldsymbol{g}_k}\right) \boldsymbol{g}_k \quad .$$

Problem: Hessian not positive definit I

Diagonalization:

$$\boldsymbol{H} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T = \sum_i \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^T \ ,$$

(vectors $\boldsymbol{u}_i$ are the eigenvectors of $\boldsymbol{H}$, and $\boldsymbol{D}$ is diagonal, with the eigenvalues λ_i of $\boldsymbol{H}$ in diagonal. At least of the eigenvalues λ_i is zero or less than zero.

1. Strategy: define modified diagonal matrix $\overline{\boldsymbol{D}}$ with elements

$$\overline{\lambda}_i = \max\left(|\lambda_i|, \delta\right) \qquad \delta > 0$$

(which requires diagonalization) and use modified positive-definite Hessian

$$\overline{\boldsymbol{H}} = \boldsymbol{U}\overline{\boldsymbol{D}}\boldsymbol{U}^T \qquad \overline{\boldsymbol{H}}^{-1} = \boldsymbol{U}\overline{\boldsymbol{D}}^{-1}\boldsymbol{U}^T$$

Problem: Hessian not positive definit II

2. Strategy: define modified Hessian by

$$\overline{\boldsymbol{H}} = \boldsymbol{H} + \alpha \boldsymbol{I}_n \qquad \text{or} \qquad \overline{\boldsymbol{H}_{jj}} = (1+\lambda)\cdot \boldsymbol{H}_{jj} \quad \text{(Levenberg-Marquardt)}$$

(simpler and faster compared to diagonalization).

- Large value of α (or λ) means search direction close to (negative) gradient (safe),
- small value of α (or λ) means search direction close to the Newton direction.

The factor α allows to interpolate between the methods of steepest descent and Newton method.

But what is large and what is small α: scale defined by

$$\alpha \approx \frac{|\boldsymbol{g}^T \boldsymbol{H} \boldsymbol{g}|}{\boldsymbol{g}^T \boldsymbol{g}}$$

suggested by step of steepest descent method.

Derivative calculation

The (optimal) Newton method requires

- first derivatives of $F(\boldsymbol{x})$: computation $\propto n$;
- second derivatives of $F(\boldsymbol{x})$: computation $\propto n^2$.

- Analytical derivatives may be impossible or difficult to obtain.
- Numerical derivatives require good step size δ for differential quotient

E.g. numerical derivative of $f(x)$ in one dimension:

$$\left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{x=x_0} \approx \frac{f(x_0+\delta)-f(x_0-\delta)}{2\delta}$$

Can the Newton (or quasi Newton) method be used without explicit calculation of the complete Hessian?

Variable metrik method

Calculation of Hessian (with $n(n+1)/2$ different elements) from sequence of first derivatives (gradients): by update of estimate $\boldsymbol{H}_k$ from change of gradient.

- Step $\boldsymbol{\Delta x}$ is calculated from $\boldsymbol{H}_k\, \boldsymbol{\Delta x} = -\boldsymbol{g}_k$
- After a line search with minimum at z_{min} new value is $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + z_{\text{min}}\, \boldsymbol{\Delta x}$,
  with gradient $\boldsymbol{g}_{k+1}$.
- Update matrix: $\boldsymbol{U}_k$ is added to $\boldsymbol{H}_k$ to get new estimate of Hessian:

$$\boldsymbol{H}_{k+1} = \boldsymbol{H}_k + \boldsymbol{U}_k$$

$$\boldsymbol{H}_{k+1}\boldsymbol{\delta} = \boldsymbol{\gamma} \qquad \boldsymbol{\delta} = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \qquad \boldsymbol{\gamma} = \boldsymbol{g}_{k+1} - \boldsymbol{g}_k$$

(with $\boldsymbol{\gamma}^T\boldsymbol{\delta} > 0$)

Note: an accurate line search is essential for the success.

Variable metrik method, contnd.

Most effective update formula (Broydon/Fletcher/Goldfarb/Shanno (BFGS))

$$\boldsymbol{H}_{k+1} = \boldsymbol{H}_k - \frac{(\boldsymbol{H}_k\boldsymbol{\delta})\,(\boldsymbol{H}_k\boldsymbol{\delta})^T}{\boldsymbol{\delta}^T\boldsymbol{H}_k\boldsymbol{\delta}} + \frac{\boldsymbol{\gamma}\boldsymbol{\gamma}^T}{\boldsymbol{\gamma}^T\boldsymbol{\delta}} \;.$$

Initial matrix may be the unit matrix.

Properties: the method generates n linear independent search directions for a quadratic function and the estimated Hessian $\boldsymbol{H}_{k+1}$ converges to the true Hessian $\boldsymbol{H}$.

Potential problems:

- no real convergence for "good" starting point;
- estimate "destroyed" for small, inaccurate steps (round-off errors).

Example: Minuit *from* Cern

Several options can be selected:

Option MIGRAD: minimizes the objective function, calculates first derivatives numerically and uses the BFGS update formula for the Hessian – fast (if it works)

Option HESSE: calculates the Hesse matrix numerically – recommended after minimization

Option MINIMIZE: minimization by MIGRAD and HESSE calculation with checks.

Restricted range of parameters possible: by transformation from a finite range (of external variables) to $-\infty \ldots +\infty$ (of internal variables) using trigonometrical functions.

- allows to use minimization algorithm without range restriction;
- good (quadratic) function is transformed perhaps to a difficult function;
- range specification may be bad, e.g $[0, 10^8]$ for optimum at 1.