Nonlinear optimization problem – iteration optimization methods

I. Local optimization technique
II. Global optimization technique

Ad.I Local optimization algorithms- for nonlinear programming problem:

- without constraints

$$\min_{\mathbf{x}\in R^n} f(\mathbf{x}) = f\left(\hat{\mathbf{x}}\right)$$

- with constraints

$$\min_{\mathbf{x}\in X} f(\mathbf{x}) = f\left(\hat{\mathbf{x}}\right)$$

The local optimization algorithms are convergent to a local minimum point x*, if such point exists.

Ad I. Local optimisation technique

Iterative optimization algorithms

- ❖**Optimization algorithms in one direction**
- ❖**Optimization algorithms without constraints**
- ❖**Optimization algorithms with constraints.**

Local optimization algorithm – searching the extreme point as concerns the function minimization in a feasible set of solutions, according to the iterative schedule.

Local optimisation algorithm without constraints

(1) Try to select the starting point **xo=xk**.
(2) Calculate the value of an objective function in this point ***f(x^k)*** and if it is necessary, determines its gradient **∇ *f(x^k)* .**

(3) Check the convergence criterium: Zbadaj przyjęte kryterium zbieżności.
•If YES– optimal solution ***x^k*** and an optimal value of an objective function ***f(x^k)*** is achieved
•If NO – Go to Step (4) .

(4) Calculate the fixed direction : d^k

(5) Calculate the directional minmum with the help of chosen method:

$$x^{k+1} \in T(x^k, d^k)$$

(6) Take $x^k \Leftarrow x^{k+1} \quad oraz \quad k \Leftarrow k+1$

Go to Step (2)

Minimisation in one direction:

Without gradient algorithms

- Golden section search,
- parabolic aproximation,

Gradient algorithms

- quadratic aproximation with a monoclinic test,
- quadratic aproximation with two clinic Golstein test.

One iteration *No k* of an algorithm, searching minimum point in one direction:

1. Evaluate the choosen direction d^k.

2. Find α **, which minimize** $\tilde{f}(\alpha) = f(x^k + \alpha d^k)$ **as concerns** α.

3. Take the new point: $x^{k+1} = x^k + \alpha d^k$.

The golden section search

The **golden section search** is a technique for finding the extremum of a strictly strictly unimodal by successively narrowing the range of values inside which the extremum is known to exist.

The technique derives its name from the fact that the algorithm maintains the function values for triples of points whose distances form a golden ratio.

$$\varphi = \frac{1+\sqrt{5}}{2} = 1.61803398\ldots$$

Convergence criteria:

1. Theoretical tests

$$\left|f(x^k)-f(\hat{x})\right|\le\varepsilon_1,\left\|x^k-\hat{x}\right\|\le\varepsilon_2$$

2. Stationarity solutions

$$\nabla f(x^k)=g^k \quad\Rightarrow\left\|g^k\right\|\le\varepsilon$$

3. Practical tests:

$$\left\|x_i^k-x_i^{k+1}\right\|\le\varepsilon_i,\quad\forall i=1,...,n$$

or

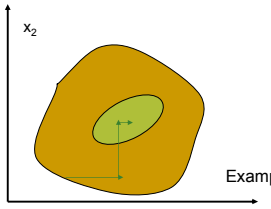
$$\left|f(x^k)-f(x^{k+1})\right|\le\varepsilon_1$$

Algorithms for local optimization

- **Non-gradient algorithms**
  - **Hooke-Jeeves algorithm**
  - **Nelder-Meade algorithm**
  - **Gauss – Seidel algorithm**
  - **Powell algorithm**
- **Gradient algorithms**
  - **Gradient descent algorithm (steepest descent algorithm)**
  - **Modified Newton algorithm**
  - **Fletcher – Reeves algorithm**
  - **Polak – Ribiery algorithm**
  - **Fletcher – Powell – Davidon algorithm**

Basic methods with directions of improvement

1. Gauss-Seidel method (without gradient).
2. The steepest descent method (gradient).
3. Newton method (gradient and hesjan).



Example of Gauss-Seidel method

$$\mathbf{d}^{(k)}=\mathbf{e}_{i^{(k)}}$$

Gauss-Seidel method (very slow linear convergence)

$$\mathbf{d}^{(k)}=-\nabla f\left(\mathbf{x}^{(k)}\right)$$

The steepest descent method (linear convergence).

$$\mathbf{d}^{(k)}=-\mathbf{H}\left(\mathbf{x}^{(k)}\right)^{-1}\nabla f\left(\mathbf{x}^{(k)}\right)$$

$$\mathbf{H}=\left\{h_{ij}\right\}=\left\{\frac{\partial^2 f}{\partial x_i\partial x_j}\right\}_{i,j\in\{1,2,...,n\}}$$

Newton method (the quadratic convergence, very expensive and sometimes not stable)

The most effective methods are named: quasi-Newton methods.

In each iteration a hesjan approximation is constructed.

Gauss – Seidel algorithm

The essence idea is an objective function minimization along each direction of the ortogonal base. This base has to be parallel to the ccordinate Kartezjan system.

Gauss - Seidel algorithm consists of using the projection T in a cyclic way to each generated direction. One cycle is named k-iteration.

Projection T:

$$T(x,d^i)=\left\{x_i^{k+1}:f(x_i^{k+1})=\min_{\tau\in\sigma}f(x_i^k+\tau_i d^i),\quad x_i^{k+1}=x_i^k+\tau_i d^i\right\}$$

Gauss – Seidel method

(1) Generate initial point $\mathbf{x^o=x^k}$. Evaluate an objective function value $\boldsymbol{f(x^k)}$

(2) Check the convergence criterium:

$$\left\|x_i^k-x_i^{k+1}\right\|\le\varepsilon_i,\quad\forall i=1,...,n \quad\text{and}\quad \left|f^k-f^{k+1}\right|\le\varepsilon_1$$

$$where\quad \varepsilon\in[0,\delta]\quad :\varepsilon=10^{-6}$$

If yes – END of algorithm, if no – Go to Step (3).

(3) Create the direction, according to the ortogonal base

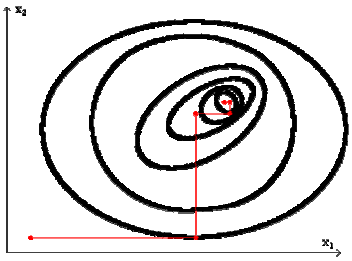
$d^k = e^k$ Np. $e^1 = [1,0,...,0]$

(4) Evaluate the minimum value of an objective function by a chosen algorithm in created direction:

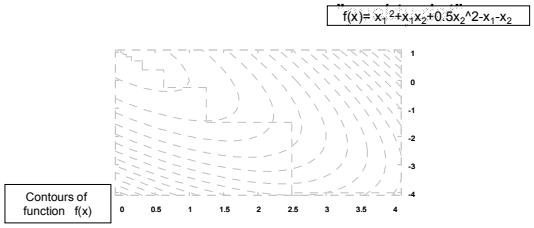
$$x^{k+1} \in T(x^k, d^k)$$

(5) Let $x^k \Leftarrow x^{k+1}$ *and* $k \Leftarrow k+1$ *and* *repeat* (1)

Illustrating scheme of a local optimisation by Gauss-Seidel algorithm on two-dimensional function contours



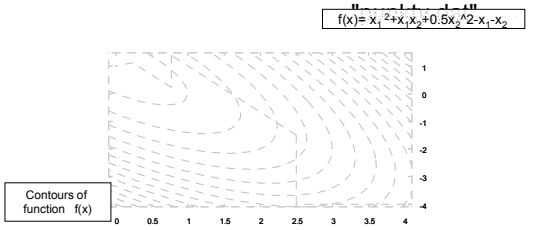
Gauss-Seidel algorithm



Powell's conjugate direction method

- Ta metoda jest stosowana dla funkcji, których poziomice mają kształt wąskich dolin.
  Można dzięki niej uzyskać znaczną poprawę szybkości zbieżności w stosunku do metody Gaussa-Seidela.
- Modyfikacja kierunku poszukiwań następuje tu w wyniku wprowadzania do bazy ortogonalnej kierunków sprzężonych do już istniejących. Stosuje się dwa warianty metody Powella.
- W pierwszym do istniejącej bazy wprowadza się kierunki sprzężone co obieg (czyli po minimalizacji wzdłuż n kierunków obowiązującej bazy), zaś w drugim wariancie następuje to po spełnieniu określonego warunku. Ponieważ kierunki wzajemnie sprzężone są liniowo niezależne, w obu wariantach metody Powella zachowany pozostaje warunek jednoznaczności przekształcenia bazy kierunków poszukiwań. Dzięki temu mamy pewność, iż nie nastąpi redukcja wymiarowości bazy, co prowadziłoby do niezbieżności metody.

Powell method



Nelder - Meade algorithm

- Let construct the simplex polygon with (n+1) tops
- Calculate a means of symmetry of simplex
- Use the following operations:
  - Reflection operator
  - Expansion operator
  - Contraction operator
- Example for NM method

The steepest descent method

It is gradient method, which allows to find the minimal value of differentiable nonlinear function f(x).

The idea of steepest descent method follow the lemma concerning the feasible direction - if there exists a direction d in space R^n such that:

$$\langle \nabla f(x), d \rangle < 0$$

then

$$f(x + \tau d) < f(x)$$

The steepest descent method

(1) Generate initial point $\mathbf{x^o=x^k}$. Evaluate an objective function value $\boldsymbol{f(x^k)}$ and its gradient $\boldsymbol{\nabla f(x^k)}$

(2) Check the convergence criterium:

$$\langle \nabla f(x^k), \nabla f(x^k) \rangle = 0 \quad then \quad \langle \nabla f(x^k), \nabla f(x^k) \rangle \le \varepsilon$$

$$where \quad \varepsilon \in [0, \delta] \quad : \varepsilon = 10^{-6}$$

If yes – END of algorithm, if no – Go to Step (3).

(3) Create the direction, according to :

$$d^k = -\nabla f(x^k)$$

(4) Evaluate the minimum value of an objective function by a chosen algorithm in a created direction:

$$x^{k+1} \in T(x^k, d^k)$$

(5) Let $x^k \Leftarrow x^{k+1}$ *and* $k \Leftarrow k+1$ *and repeat* (1)

In the minimization proces in chosen direction – the gradient bisection algorithm with two oblique Golstein test is used:

Example for the function:

$$f(x_1, x_2) = (x_1)^2 + 2(x_2)^2 - 6x_1 + x_1x_2$$

- Initial point $x^0 = [0, 0]^T$
- Direction $d = [1, 0]^T$
- Test coefficient $\beta = \frac{2}{5}$
- Initial value of the step $\tau_R = 9$
- Precision $\varepsilon' = 10^{-5}$

Derivative in the fixed direction: $p = \nabla f(x^o)^T d$

$$\nabla f(x) = \left[\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}\right] = \left[2x_1 - 6 + x_2, 4x_2 + x_1\right]$$

for $x = x^0 = [0, 0]^T$

$$\nabla f(x^0) = [-6, 0]$$

and:

$$p = \nabla f(x^o)^T d = [-6 \quad 0] \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = -6$$

(2) Calculate $\tau = \frac{1}{2}(\tau_L + \tau_R)$ and $f(x^0 + \tau d)$.

$$\tau = \frac{1}{2}(\tau_R) = \frac{1}{2}(9) = 4{,}5$$

$$f(x^0 + \tau d) = f\left[(0{,}0) + (4{,}5 \quad ,0)\right] = 20{,}25 - 27 = -6{,}75$$

Go to Step (3)

(3) If $f(x^0 + \tau d) < f(x^0) + (1-\beta) p\tau$

Then take $\tau_{\cdot} \Rightarrow \tau$

And go to Step (2). Otherwise go to Step (4)

Try to control the condition : -6,75 <?

$$0 + (-6) \cdot (0{,}6) \cdot (4{,}5) = -16{,}2$$

No

Go to Step (4)

(4) If $f(x^{\circ} + \tau d) > f(x^{\circ}) + \beta p\tau$

then $\tau_R \Rightarrow \tau$

And go to Step (2). Otherwise End of the algorithm

Try to control the condition: -6,75 >?

$$0 + (-6) \cdot (0{,}4) \cdot (4{,}5) = -10{,}8$$

Yes

and go to Step (2)

Second Iteration

(...)

After the third iteration the result is as follows: $\tau = 3{,}375$

Example of iteration steepest descent algorithm for the following function:

$f(x_1, x_2) = 2(x_1)^2 + (x_2)^2 - 2\, x_1 x_2$

- Initial point $x^0 = [2, 3]^T$
- Test coefficient $\beta = \frac{1}{4}$
- Initial value of step $\tau_R = 1$

- Calculate $d^0 = -\nabla f(x^0) = [-2, -2]^T$
- The first value of step $\tau_R = 1$ fulfills the Golstein test:
- $x^1 = x^0 + \tau_0 d^0 = [0\ 1]^T$
- In second iteration: $d^1 = -\nabla f(x^1) = [2 \quad -2]^T$
- We receive: $f(x^1 + \tau d^1) = 20\tau^2 - 8\tau + 1$

$$p = \nabla f(x^1)^T d^1 = \begin{bmatrix} -2 \\ 2 \end{bmatrix} \cdot [2 \quad -2] = -8$$

- Golstein test has a form:
- $-6 \le 20\tau^2 - 8\tau \le -2$
- In third iteration we calculate $\tau_1 = 0{,}25$

So

- $x^2 = x^1 + \tau_1 d^1 = [\frac{1}{2} \quad \frac{1}{2}]^T$
- According the discussed algorithm we calculate the following iterations till the termination condition will be fulfilled.

The steepest descent algorithm for the following function:

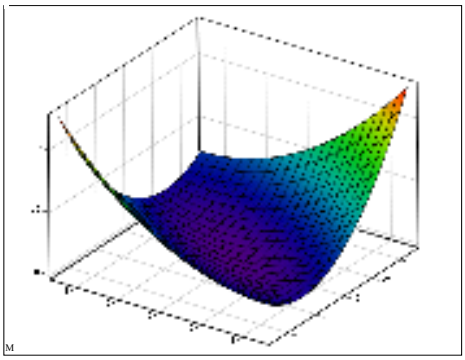
- $F(x_1, x_2) = 2(x_1)^2 + (x_2)^2 - 2\,x_1x_2$
- $x^0 = [2\ \ 3]$
- $x^1 = [0\ \ 1]$
- $x^2 = [\frac{1}{4}\ \ \frac{1}{2}]$
- $x^3 = [\frac{1}{4}\ \ \frac{1}{4}]$
- $x^4 = [\quad\ \ ]$ etc. ....

- Till the termination condition will be fulfilled:

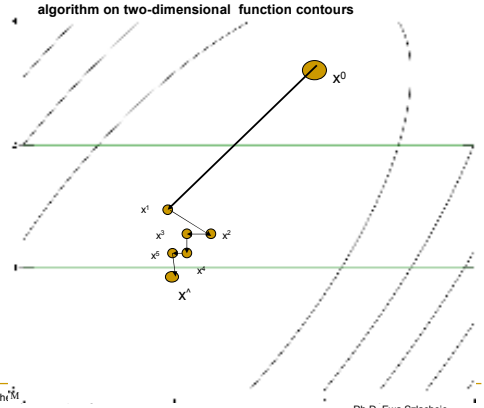
$$\left\langle \nabla f(\hat{x}), \nabla f(\hat{x}) \right\rangle < \varepsilon = 10^{-3}$$

The optimal solution $x^k=[0,0]$ and $f(x^k)=0$.

Objective function f(x)



Illustrating scheme of a local optimisation by the steepest descent algorithm on two-dimensional function contours



Newton algorithm

(1) Generate initial point $\mathbf{x^o=x^k}$. Evaluate an objective function value $f(x^k)$ and its gradient $\nabla f(x^k)$

(2) Check the convergence criterium:

$$\left\langle \nabla f(x^k), \nabla f(x^k) \right\rangle = 0 \ \Rightarrow\ \left\langle \nabla f(x^k), \nabla f(x^k) \right\rangle \le \varepsilon$$

$$where \quad \varepsilon \in [0, \delta] \quad : \varepsilon = 10^{-6}$$

If yes – END of algorithm, if no – Go to Step (3).

(3) Create the direction, according to :

$$d^k = -H^{-1}\nabla f(x^k)$$

(4) Evaluate the minimum value of an objective function by a chosen algorithm in a created direction:

$$x^{k+1} \in T(x^k, d^k)$$

(5) Let $x^k \Leftarrow x^{k+1}$ *and* $k \Leftarrow k+1$ *and repeat* *Step* (1)

The definition of direction for gradient optimization methods:

- 1. **Polak – Ribiery method:**

  Calculate conjugate direction $d^{k+1} = -\nabla f(x^{k+1}) + \beta_k d^k$

  where: $\beta_k = \dfrac{\left\langle (\nabla f(x^{k+1}) - \nabla f(x^k)), \nabla f(x^{k+1}) \right\rangle}{\left\langle \nabla f(x^k), \nabla f(x^k) \right\rangle}$

- 2. **Fletcher – Reeves method**

  Calculate conjugate direction $d^{k+1} = -\nabla f(x^{k+1}) + \beta_k d^k$

  where: $\beta_k = \dfrac{\left\langle \nabla f(x^{k+1}), \nabla f(x^{k+1}) \right\rangle}{\left\langle \nabla f(x^k), \nabla f(x^k) \right\rangle}$

- 3. **Davidon-Fletcher-Powell method (DFP)**

DFP – the modification of matrix V_k,

V_k has to be equal the matrix H^{-1} for

$k \to \infty$

$$d^k = -V_k \nabla f(x^k)$$

$$x^{k+1} = x^k + \beta_k d^k$$

$$V_{k+1} = V_k + \frac{d^k >< d^k}{\left\langle d^k, Ad^k \right\rangle}$$