Algoritmi numerici pentru optimizare

III - Algoritmi determinişti de ordin superior

Prof.dr.ing. Gabriela Ciuprina

Universitatea "Politehnica" Bucureşti, Facultatea de Inginerie Electrică,
Departamentul de Electrotehnică

Suport didactic pentru disciplina *Algoritmi Numerici*, 2017-2018

Cuprins

Formularea problemei

Să se găsească n parametri independenţi, notaţi x_1^*, x_2^*, $\ldots$, x_n^*, pentru care expresia E este minimă, unde

$$E = f(x_1, x_2, \ldots, x_n), \tag{1}$$

şi $f : \Omega \to \mathbb{R}$, $\Omega \subset \mathbb{R}^n$, este dată.

Pe scurt:

$$(x_1^*, x_2^*, \ldots, x_n^*) = \arg\min f(x_1, x_2, \ldots, x_n). \tag{2}$$

Notaţii

$$\mathbf{x} = [x_1, x_2, \ldots, x_n]^T \in \Omega. \tag{3}$$

$$\mathbf{x}_{\min} = [x_1^*, x_2^*, \ldots, x_n^*]^T \in \Omega \tag{4}$$

$$\mathbf{x}_{\min} = \arg\min f(\mathbf{x}), \quad \mathbf{x} \in \Omega$$

$$E_{\min} = f(\mathbf{x}_{\min}).$$

Minime globale/locale

$$\mathbf{x}_{\min} = \arg\min f(\mathbf{x}), \quad \mathbf{x} \in \Omega \tag{5}$$

$$E_{\min} = \min\{\, f(\mathbf{x}) |\, \mathbf{x} \in \Omega\} \tag{6}$$

$\mathbf{x}_{\text{min}}$ este *minim global* dacă

$$E_{\min} \le f(\mathbf{x}), \quad \forall\, \mathbf{x} \in \Omega \tag{7}$$

- dacă $E_{\text{min}} \le f(\mathbf{x})$ doar într-o vecinătate a lui $\mathbf{x}_{\text{min}}$ atunci minimul este *local*.
- în practică este dificil de stabilit dacă un minim găsit este local sau global;
- minimul global s-ar putea să nu fie unic.

Metode de optimizare

I. Deterministe - conduc la aceeaşi soluţie pentru rulări diferite ale programului, dacă pornesc din aceleaşi condiţii iniţiale şi au aceiaşi parametri.

- Dezavantaj: ?
- Avantaj: ?

Pot fi

1. de ordin zero
2. de ordin superior (1,2) Ex: metoda Newton; metoda falsei poziţii; metoda gradientului (a celei mai rapide coborâri); metoda gradienţilor conjugaţi; metode cvasi-Newton, etc.

II. Stocastice

Metode de optimizare

$$(x_1^*, x_2^*, \ldots, x_n^*) = \arg\min f(x_1, x_2, \ldots, x_n). \tag{8}$$

"Optimizare 1D": $n = 1$
"Optimizare nD": $n > 1$

Metode

- *de căutare* = intervalul care conţine minimul este micşorat prin evaluarea lui f în anumite puncte;
- *de aproximare* = funcţia de optimizat este aproximată printr-o funcţie cunoscută care poate fi analizată uşor.

Metode de aproximare

Ideea: se aproximează local relieful funcţiei cu o funcţie polinominală de grad mic, căreia i se poate determina uşor minimul.

Metodele se numesc:

- de ordinul 1 - dacă necesită evaluarea primei derivate a funcţiei obiectiv;
- de ordinul 2 - dacă necesită evaluarea celei de a doua derivate a funcţiei obiectiv;
- etc.

Condiţii necesare şi suficiente - 1D

Fie $f : \mathbb{R} \to \mathbb{R}$ cu derivate de ordinul 1 şi 2 continue.

Condiţia necesară de minim

Dacă x^* este un minim local al lui f atunci el este un punct critic pentru f:

$$\frac{\mathrm{d}f}{\mathrm{d}x}(x^*) = 0.$$

Condiţia suficientă de minim

Dacă

$$\frac{\mathrm{d}f}{\mathrm{d}x}(x^*) = 0 \quad \text{şi} \quad \frac{\mathrm{d}^2 f}{\mathrm{d}x^2}(x^*) > 0.$$

atunci x^* este un punct de minim local pentru f,

Obs:

- Dacă $\mathrm{d}^2 f/\mathrm{d}x^2(x^*) < 0$ atunci punctul este de maxim.
- Dacă $\mathrm{d}^2 f/\mathrm{d}x^2(x^*) = 0$ testul derivatei a doua nu este concludent şi trebuie folosită dezvoltarea în serie Taylor pentru a analiza funcţia.

Metoda aproximării parabolice

Ideea:

- se aproximează funcţia de minimizat $f(x)$ în jurul punctului de minim cu o parabolă $q(x)$ şi se alege minimul parabolei ca fiind o aproximare pentru minimul căutat;
- procedeul continuă iterativ până când derivata funcţiei este neglijabilă în minimul aproximativ.

Parabola $q(x)$ e determinată de trei puncte $\Rightarrow$ este nevoie de trei informaţii
$\Rightarrow$ există mai multe variante

Metoda aproximării parabolice

a) Metoda Newton

Cele trei informaţii sunt legate de acelaşi punct x_k (valoarea funcţiei, derivatei de ordinul 1 şi a celei de ordinul 2).
Condiţii impuse:

$$q(x_k) = f(x_k), \tag{9}$$
$$q'(x_k) = f'(x_k), \tag{10}$$
$$q''(x_k) = f''(x_k). \tag{11}$$

$\Rightarrow$

$$q(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{f''(x_k)}{2}(x - x_k)^2. \tag{12}$$

Metoda aproximării parabolice

a) Metoda Newton

$$q(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{f''(x_k)}{2}(x - x_k)^2. \qquad (13)$$

Estimarea x_{k+1} se obţine din condiţia $q'(x_k + 1) = 0$, unde

$$q'(x) = f'(x_k) + f''(x_k)(x - x_k)$$

$$f'(x_k) + f''(x_k)(x_{k+1} - x_k) = 0$$

$\Rightarrow$

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)} \qquad (14)$$

Metoda aproximării parabolice

a) Metoda Newton

- Noua iteraţie nu depinde de valoarea funcţiei în x_k;
- Este o metodă de ordinul doi;
- Metoda Newton a fost prezentată şi ca metodă de rezolvare a ecuaţiilor neliniare.

$$g(x) = 0 \quad \Leftrightarrow \quad \min f(x)$$

dacă $g(x) = f'(x)$.

$$x_{k+1} = x_k - \frac{g(x_k)}{g'(x_k)} \quad \Leftrightarrow \quad x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

Metoda aproximării parabolice

b) Metoda falsei poziţii

Condiţii impuse:

$$
\begin{aligned}
q(x_k) &= f(x_k), & (15)\\
q'(x_k) &= f'(x_k), & (16)\\
q'(x_{k-1}) &= f'(x_{k-1}). & (17)
\end{aligned}
$$

$\Rightarrow$

$$
\begin{aligned}
q(x) &= f(x_k) + f'(x_k)(x - x_k) + c(x - x_k)^2 & (18)\\
q'(x) &= f'(x_k) + 2c(x - x_k) & (19)
\end{aligned}
$$

c se determină impunând (17).

Metoda aproximării parabolice

$\Rightarrow$

$$c = \frac{f'(x_{k-1} - f'(x_k))}{2(x_{k-1} - x_k)}$$

$\Rightarrow$

$$q(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{f'(x_{k-1} - f'(x_k))}{(x_{k-1} - x_k)} \frac{(x - x_k)^2}{2} \quad (20)$$

Comparând cu Newton - derivata a doua $\approx$ formulă de diferenţe finite, de ordinul 1.

$g'(x) = 0 \Rightarrow$

$$x_{k+1} = x_k - f'(x_k) \frac{x_{k-1} - x_k}{f'(x_{k-1}) - f'(x_k)} \quad (21)$$

Metoda aproximării parabolice

b) Metoda falsei poziţii

- Noua iteraţie nu depinde de valoarea funcţiei în x_k;
- Este o metodă de ordinul unu;
- Metoda falsei poziţii pentru minimizare este metoda secantei pentru rezolvarea unei ecuaţii neliniare.

$$g(x) = 0 \quad \Leftrightarrow \quad \min f(x)$$

dacă $g(x) = f'(x)$.

$$x_{k+1} = x_k - g(x_k)\frac{x_k - x_{k-1}}{g(x_k) - g(x_{k-1})} \quad \Leftrightarrow \quad x_{k+1} = x_k - f'(x_k)\frac{x_{k-1} - x_k}{f'(x_{k-1}) - f'(x_k)}$$

Metoda aproximării parabolice

c) Metoda interpolării parabolice

Condiţii impuse:

$$q(x_k) = f(x_k), \tag{22}$$
$$q(x_{k-1}) = f(x_{k-1}), \tag{23}$$
$$q(x_{k-2}) = f(x_{k-2}) \tag{24}$$

$\Rightarrow$

$$\begin{aligned} q(x) &= \frac{(x-x_{k-1})(x-x_{k-2})}{(x_k-x_{k-1})(x_k-x_{k-2})} f(x_k) + + \\ &+ \frac{(x-x_k)(x-x_{k-2})}{(x_{k-1}-x_k)(x_{k-1}-x_{k-2})} f(x_{k-1}) + \\ &+ \frac{(x-x_k)(x-x_{k-1})}{(x_{k-2}-x_k)(x_{k-2}-x_{k-1})} f(x_{k-2}) \end{aligned} \tag{25}$$

$q'(x) = 0 \Rightarrow x_{k+1}$ Metode de ordin zero

Metoda aproximării parabolice

Observaţii:

- Iniţializarea este simpla la Newton, dublă la falsa poziţie şi triplă ă interpolarea polinomială;
- Ordinul metodei de optimizare este 2 la Newton, 1 la falsa poziţie şi 0 la iniţializarea polinomială.
- La orice metodă estimările pot conduce catre un punct staţionar (maxim sau de inflexiune) nedorit, de aceea algoritmul trebuie completat cu paşi suplimentari de testare şi corectare.
- Toate aceste metode sunt exacte pentru funcţii parabolice.
- Efortul de calcul pe iteraţie: Newton (o derivata de ordin 1 şi una de ordin 2), falsa poziţie (o derivată de ordin 1), interpolarea parabolică (o evaluare de funcţie).

Metoda aproximării parabolice - algoritm

funcţie [**real** x_min,f_min, **întreg** n_iter] = opt_falsa_pozitie(**real** x_1, x_2,
funcţie f, **funcţie** f_der, **întreg** NMAX, **real** EPS)

$\mathrm{stopval} = 1.e + 4$
$\mathrm{fdvec}(1) = f\_\mathrm{der}(x_1)$
$x(1) = x_1$
$\mathrm{fdvec}(2) = f\_\mathrm{der}(x_2)$
$x(2) = x_2$
$k = 2$
cât timp ($k \le$NMAX **şi** $\mathrm{stopval} \ge$ EPS)
 $x(k+1) = x(k) - \mathrm{fdvec}(k) * (x(k-1) - x(k))/(\mathrm{fdvec}(k-1) - \mathrm{fdvec}(k))$
 $\mathrm{fdvec}(k+1) = f\_\mathrm{der}(x(k+1))$
 $\mathrm{stopval} = |\mathrm{fdvec}(k+1)|$
 $k = k + 1$
$x\_\min = x(k)$
$f\_\min = f(x(k))$
$n\_\mathrm{iter} = k - 2$
retur

Metoda aproximării cubice

Aproximarea locală se face cu un polinom de gradul 3.
Condiţii impuse:

$$\begin{aligned} q(x_k) &= f(x_k), & (26)\\ q'(x_k) &= f'(x_k), & (27)\\ q(x_{k-1}) &= f(x_{k-1}), & (28)\\ q'(x_{k-1}) &= f'(x_{k-1}) & (29) \end{aligned}$$

$\Rightarrow$

$$x_{k+1} = x_k - (x_k - x_{k-1}) \left[\frac{f'(x_k) + u_2 - u_1}{f'(x_k) - f'(x_{k-1}) + 2u_2} \right] \quad (30)$$

unde

$$\begin{aligned} u_1 &= f'(x_{k-1}) + f'(x_k) - 3\frac{f(x_{k-1}) - f(x_k)}{x_{k-1} - x_k}, & (31)\\ u_2 &= \sqrt{u_1^2 - f'(x_{k-1})f'(x_k)}. & (32) \end{aligned}$$

Condiţii necesare şi suficiente - nD

Fie $f : \mathbb{R}^n \rightarrow \mathbb{R}$ cu derivate de ordinul 1 şi 2 continue.

Condiţia necesară de minim

Dacă $\mathbf{x}^*$ este un minim local al lui f atunci el este un punct critic pentru f:

$$\frac{\partial f}{\partial x_j}(\mathbf{x}^*) = 0, \quad \forall j = 1, \dots, n.$$

$\Leftrightarrow$ vectorul gradient să fie nul.

$$\nabla f(\mathbf{x}^*) = \mathbf{0} \tag{33}$$

Condiţia suficientă de minim

Matricea Hessian să fie pozitiv definită în punctul critic

$$\nabla^2 f(\mathbf{x}^*) > 0 \tag{34}$$

Metoda Newton

Taylor:

$$f(\mathbf{x}) = f(\mathbf{x}_k) + (\mathbf{x} - \mathbf{x}_k)^T \nabla f(\mathbf{x}_k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^T \nabla^2 f(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k) + \cdots \quad (35)$$

unde $\mathbf{x}, \mathbf{x}_k \in \mathbb{R}^n$.

$$f(\mathbf{x}) \approx g(\mathbf{x})$$

unde

$$g(\mathbf{x}) = f(\mathbf{x}_k) + (\mathbf{x} - \mathbf{x}_k)^T \mathbf{g}_k + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^T \mathbf{H}_k (\mathbf{x} - \mathbf{x}_k) \quad (36)$$

unde am notat $\mathbf{f}_k = f(\mathbf{x}_k)$, $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$, $\mathbf{H}_k = \nabla^2 f(\mathbf{x}_k)$

Metoda Newton

$$q(\mathbf{x}) = f(\mathbf{x}_k) + (\mathbf{x} - \mathbf{x}_k)^T \mathbf{g}_k + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^T \mathbf{H}_k (\mathbf{x} - \mathbf{x}_k) \qquad (37)$$

Se impune condiţia

$$\nabla q(\mathbf{x}) = 0 \quad \mathbf{x}_{k+1}$$

$$\nabla q(\mathbf{x}) = \mathbf{g}_k + \mathbf{H}_k (\mathbf{x} - \mathbf{x}_k)$$

$\Rightarrow$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}_k^{-1} \mathbf{g}_k \qquad (38)$$

Metoda Newton

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}_k^{-1}\mathbf{g}_k \tag{39}$$

Observaţii:

- Relaţia este similară cu cea de la cazul 1D;
- În practică nu se inversează matricea Hessian, ci se rezolvă

$$\mathbf{H}_k\mathbf{p}_k = -\mathbf{g}_k \quad \mathbf{p}_k \tag{40}$$

  apoi se calculeaa

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k \tag{41}$$

- Dezavantaj: efort mare de calcul (o matrice Hessian la fiecare iteraţie şi o rezolvare de sistem)

Metoda gradientului

Am mai discutat această metodă şi la rezolvarea sistemelor simetrice şi pozitiv definite. Pe scurt:

$$\mathbf{A}(\mathbf{x}) = \mathbf{b} \quad \Leftrightarrow \quad \min f(x) = \frac{1}{2}\mathbf{x}^T\mathbf{A} - \mathbf{b}^T\mathbf{x} + \mathbf{c}$$

Deoarece

$$\nabla f = \frac{1}{2}(\mathbf{A}^T + \mathbf{A}) - \mathbf{b} = \mathbf{A}\mathbf{x} - \mathbf{b}$$

Algoritmul se baza pe

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{r}^{(k)}$$

unde $\mathbf{r}^{(k)} = -\nabla f(\mathbf{x}^{(k)})$
α_k se determina din $t(\alpha_k) = f(\mathbf{x}^{(k+1)})$ min, de unde

$$\alpha_k = \frac{(\mathbf{r}^{(k)})^T(\mathbf{r}^{(k)})}{(\mathbf{r}^{(k)})^T\mathbf{A}\mathbf{r}^{(k)}}$$

Metoda gradientului

Acum pornim direct cu optimizarea funcţiei neliniare $f(\mathbf{x})$
Ideea: $f(\mathbf{x}) \approx l(\mathbf{x})$ (aproximarea de ordinul 1)[1]

$$l(\mathbf{x}) = f_k + (\mathbf{x} - \mathbf{x}_k)^T \mathbf{g}_k$$

La o nouă iteraţie $\mathbf{x}_{k+1}$, valoarea funcţiei

$$f_{k+1} \approx f_k + (\mathbf{x}_{k+1} - \mathbf{x}_k)^T \mathbf{g}_k$$

Deci $\Delta f \approx \Delta \mathbf{x}^T \mathbf{g}_k$.
Doresc o scădere maximă $\Rightarrow$ o valoare maximă a produsului scalar dintre $\Delta \mathbf{x}$ şi $\mathbf{g}_k$ $\Rightarrow$ $\Delta \mathbf{x}$ trebuie să aibă direcţia vectorului gradient şi sens opus acestuia.

Notăm

$$\Delta \mathbf{x} = -\alpha \mathbf{g} = \alpha \mathbf{v}$$

unde $\alpha \in \mathbb{R}_+$, $\mathbf{v}$ - direcţia de căutare.

[1] Notăm iteraţia ca indice.

Metoda gradientului

În concluzie ideea metodei gradientului se bazează pe construcţia

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{v}_k$$

unde

$$\mathbf{v}_k = -\mathbf{g}_k = -\nabla f(\mathbf{x}_k)$$

iar α_k se determină la fiecare iteraţie a.î. să se minimizeze funcţia f după direcţia $\mathbf{v}_k$ care trece prin $\mathbf{x}_k$, adică funcţia 1D $t(\alpha)$

$$\min t(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{v}_k) \quad \Rightarrow \quad \alpha_k$$

Se pot aplica metode de minimizare 1D.

Căutarea valorii optime pentru α

- Dacă minimizarea liniară este exactă, atunci direcţiile consecutive din metoda gradientului sunt perpendiculare

$$\mathbf{g}_{k+1}^T \mathbf{v}_k = \mathbf{v}_k^T \mathbf{g}_{k+1} = 0. \tag{42}$$

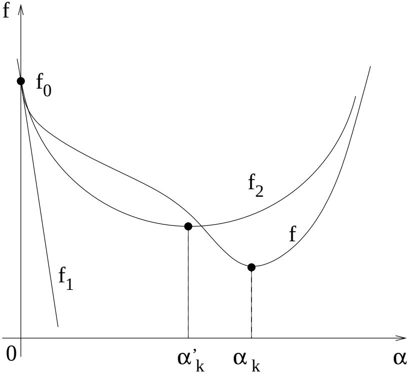
- Căutarea liniară a lui α_k s-ar putea elimina dacă se cunoaşte Hessianul H

$$f(\mathbf{g}_k + \alpha \mathbf{v}_k) \approx f(\mathbf{x}_k) + (\alpha \mathbf{v}_k)^T \mathbf{g}_k + \frac{1}{2} (\alpha \mathbf{v}_k)^T \mathbf{H}_k (\alpha \mathbf{v}_k)$$

$$t(\alpha) \approx q(\alpha) \quad q(\alpha) = f(\mathbf{x}_k) - \alpha \mathbf{g}_k^T \mathbf{g}_k + \frac{1}{2} \alpha^2 \mathbf{g}_k^T \mathbf{H}_k \mathbf{g}_k$$

$$q'(\alpha) = 0 \quad \Rightarrow \quad \alpha_k' = -\frac{\mathbf{v}_k^{\mathrm{T}} \mathbf{g}_k}{\mathbf{g}_k^{\mathrm{T}} \mathbf{H}_k \mathbf{g}_k} = \frac{\mathbf{g}_k^{\mathrm{T}} \mathbf{g}_k}{\mathbf{g}_k^{\mathrm{T}} \mathbf{H}_k \mathbf{g}_k} \tag{43}$$

Căutarea valorii optime pentru α



Funcţia *f* şi aproximările sale pe direcţia $\mathbf{v}_k$.

α_k - prin căutare liniară.

α'_k obţinută cu (43)

Care este mai bună ?

Condiţia de oprire

Variante:

- funcţie

$$\frac{|f_{k+1} - f_k|}{1/2(|f_{k+1}| + |f_k| + \text{EPS})} < \text{ftol}. \tag{44}$$

  EPS - zeroul maşinii), introdus pentru a evita împărţirea la zero.

- pas

$$\frac{\|\mathbf{x}_{k+1} - \mathbf{x}_k\|}{\|\mathbf{x}_1 - \mathbf{x}_0\|} < \text{xtol}. \tag{45}$$

- gradient

$$\|\mathbf{g}_{k+1}\| < \text{gtol}. \tag{46}$$

Algoritmul general al metodei gradientului

1. Alege $\mathbf{x}_0$
   $k = 0$
   Calculează $\mathbf{g}_0 = \nabla f(x_0)$
2. **repetă**
   2.1. $\mathbf{v}_k = -\mathbf{g}_k$
   2.2. Minimizează $t(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{v}_k)$ în raport cu $\alpha \Rightarrow \alpha_k$
   2.3. $\mathbf{p}_k = \alpha_k \mathbf{v}_k$
   2.4. $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$
   2.5. Calculează $\mathbf{g}_{k+1} = \nabla f(x_{k+1})$
   2.6 $k = k + 1$

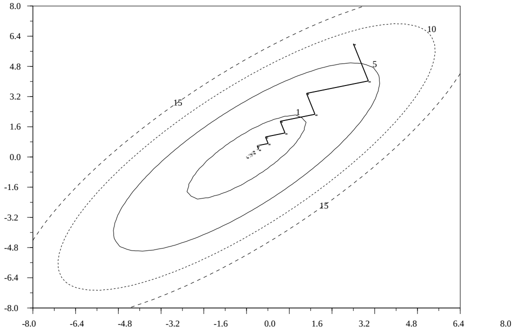
   până când este îndeplinită condiţia de oprire

În capitolul de rezolvări de sisteme liniare, algoritmul era particularizat pentru funcţii pătratice

$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \mathbf{A}\mathbf{x} - \mathbf{b}^T \mathbf{x} + \mathbf{c}$

Efortul de calcul

Efort de calcul mare - chiar şi pentru o funcţie pătratică



Metoda gradienţilor conjugaţi

Direcţia de căutare depinde atât de direcţia gradientului, cât şi de direcţia de căutare anterioară[2]

$$\mathbf{v}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{v}_k, \quad k > 0, \tag{47}$$

Primul pas este ca la metoda gradientului:

$$\mathbf{v}_0 = -\mathbf{g}_0$$

Direcţiile $\mathbf{v}_k$ sunt alese să fie H-conjugate (sau H-ortogonale):

$$\mathbf{v}_{k+1}^{\mathrm{T}} \mathbf{H}_k \mathbf{v}_j = 0, \quad (\forall)\ j \leq k. \tag{48}$$

[2] Revedeţi şi prezentarea de la sisteme algebrice liniare.

Metoda gradienţilor conjugaţi

Determinăm β înlocuind (47) în (48).

$$\mathbf{v}_{k+1}^{\mathrm{T}}\mathbf{H}_k\mathbf{v}_j = 0, \quad (\forall)\ j \leq k.$$

$$j = k \quad \Rightarrow \quad \mathbf{v}_{k+1}^{\mathrm{T}}\mathbf{H}_k\mathbf{v}_k = 0$$

$$(-\mathbf{g}_{k+1} + \beta_k\mathbf{v}_k)^T\mathbf{H}_k\mathbf{v}_k = 0$$

$\Rightarrow$

$$\beta_k = \frac{\mathbf{g}_{k+1}^{\mathrm{T}}\mathbf{H}_k\mathbf{v}_k}{\mathbf{v}_k^{\mathrm{T}}\mathbf{H}_k\mathbf{v}_k}, \tag{49}$$

Necesită calculul Hessianului la fiecare iteraţie - foarte costisitor, se procedează altfel:

Metoda gradienţilor conjugaţi

Folosind aproximaţia de ordinul doi pentru funcţia f

$$f(\mathbf{x}) \approx q(\mathbf{x}), \quad q(\mathbf{x}) = f_k + (\mathbf{x} - \mathbf{x}_k)^T \mathbf{g}_k + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^T \mathbf{H}_k (\mathbf{x} - \mathbf{x}_k)$$

$\Rightarrow$

$$\nabla f(\mathbf{x}) \approx \nabla q(\mathbf{x}), \quad \nabla q(\mathbf{x}) = \mathbf{g}_k + \mathbf{H}_k(\mathbf{x} - \mathbf{x}_k)$$

$\Rightarrow$ pentru $\mathbf{x} = \mathbf{x}_{k+1}$

$$\mathbf{g}_{k+1} \approx \mathbf{g}_k + \mathbf{H}_k(\mathbf{x}_{k+1} - \mathbf{x}_k) \quad \Rightarrow \quad \mathbf{g}_{k+1} \approx \mathbf{g}_k + \mathbf{H}_k(\alpha_k \mathbf{v}_k)$$

care înlocuită în (49) conduce la

$$\begin{aligned} \beta_k &= \frac{\mathbf{g}_{k+1}^{\mathrm{T}}(\mathbf{g}_{k+1} - \mathbf{g}_k)}{\mathbf{v}_k^{\mathrm{T}}(\mathbf{g}_{k+1} - \mathbf{g}_k)} \\ &= \frac{\mathbf{g}_{k+1}^{\mathrm{T}}(\mathbf{g}_{k+1} - \mathbf{g}_k)}{(-\mathbf{g}_k + \beta_{k-1}\mathbf{v}_{k-1})^{\mathrm{T}}(\mathbf{g}_{k+1} - \mathbf{g}_k)}. \end{aligned} \tag{50}$$

Metoda gradienţilor conjugaţi

Se poate arăta că şirul $\mathbf{x}_k$ este astfel construit încât gradientul în $\mathbf{x}_k$ şi direcţia $\mathbf{v}_k$ satisfac în plus relaţiile de ortogonalitate:

$$\mathbf{g}_{k+1}^{\mathrm{T}} \mathbf{g}_j = 0, \quad \mathbf{v}_{k+1}^{\mathrm{T}} \mathbf{g}_j = 0, \quad j \leq k, \tag{51}$$

Relaţii care înlocuite în (50) $\Rightarrow$

$$\beta_k = \frac{\mathbf{g}_{k+1}^{\mathrm{T}} (\mathbf{g}_{k+1} - \mathbf{g}_k)}{\mathbf{g}_k^{\mathrm{T}} \mathbf{g}_k} = \frac{\mathbf{g}_{k+1}^{\mathrm{T}} \mathbf{g}_{k+1}}{\mathbf{g}_k^{\mathrm{T}} \mathbf{g}_k}. \tag{52}$$

Metoda gradienţilor conjugaţi

Oricare din cele două formule din relaţia (52) se pot folosi. Fiecare din ele este cunoscută sub un nume celebru, şi anume:

$$\beta_k = \frac{\mathbf{g}_{k+1}^{\mathrm{T}} \mathbf{g}_{k+1}}{\mathbf{g}_k^{\mathrm{T}} \mathbf{g}_k} \quad \text{– formula Fletcher-Reeves} \tag{53}$$

$$\beta_k = \frac{(\mathbf{g}_{k+1} - \mathbf{g}_k)^{\mathrm{T}} \mathbf{g}_{k+1}}{\mathbf{g}_k^{\mathrm{T}} \mathbf{g}_k} \quad \text{– formula Polak-Ribiere.} \tag{54}$$

Relaţiile sunt echivalente într-o aritmetică exactă, dar numeric formula PR se preferă pentru că ea poate compensa micile "defecte" de ortogonalitate ale direcţiilor.

Algoritmul general al metodei GC

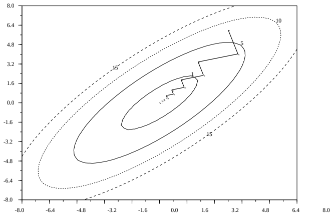
Varianta Polak-Ribiere

1. Alege $\mathbf{x}_0$
   $k = 0$
   Calculează $\mathbf{g}_0 \nabla f(\mathbf{x}_0)$
   $\beta = 0$
2. **repetă**
   2.1. $\mathbf{v}_k = -\mathbf{g}_k + \beta \mathbf{v}_{k-1}$
   2.2. Minimizează $t(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{v}_k)$ în raport cu $\alpha \Rightarrow \alpha_k$
   2.3. $\mathbf{p}_k = \alpha_k \mathbf{v}_k$
   2.4. $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$
   2.5. Calculează $\mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1})$
   2.6 $\beta = (\mathbf{g}_{k+1} - \mathbf{g}_k)^T * \mathbf{g}_{k+1} / (\mathbf{g}_k^T * \mathbf{g}_k)$
   2.7. $k = k + 1$

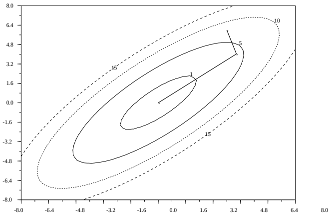
   până când este îndeplinită condiţia de oprire

Efortul de calcul

Pentru o funcţie pătratică de *n* variabile se demonstrează că GC obţine minimul după *n* paşi (într-o aritmetică exactă). $\Rightarrow$ este o metodă semiiterativă.



Metoda gradientului



Metoda gradienţilor conjugaţi

GC este în general mai rapid convergentă decât metoda gradientului pentru funcţii multidimensionale mai complexe (funcţii neparabolice).

Convergenţă

Def. Un algoritm are ordinul de convergenţă p dacă

$$\lim_{k\to\infty} \frac{\|\mathbf{x}_{k+1} - \mathbf{x}_{\min}\|}{\|\mathbf{x}_k - \mathbf{x}_{\min}\|^p} = C$$

unde C este o constantă, *rata de convergenţă*

- $p = 2$ - convergenţă pătratică(metoda Newton)
- $p = 1$, $C \neq 0$, $C < 1$ - convergenţă liniară (metoda gradientului)
- $p = 1$, $C = 0$ - convergenţă superliniară (metoda gradienţilor conjugaţi după executarea a n paşi)

Metode cvasi-Newton

Ideea:

- Simulează iteraţii de tip Newton, plasându-se între metoda gradienului şi metoda Newton. Sunt metode de ordinul 1.
- Se lucrează cu aproximări ale inversei matricei Hessian, calculată cu ajutorul vectorului gradient evaluat în iteraţiile precedente.
- Variante
  - mai simple - în care matricea rămâne constantă pe parcursul iteraţiilor;
  - mai avansate - în care se construiesc aproximări din ce în ce mai bune - algoritmi de metrică variabilă

Metoda Newton modificată

Newton: $\mathbf{x}_{k+1} - \mathbf{x}_k = -\mathbf{H}_k^{-1}\mathbf{g}_k$

Newton modificată

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{S}_k \mathbf{g}_k, \tag{55}$$

unde, $\mathbf{S}_k$ este o matrice simetrică de dimensiune $n \times n$
$\alpha_k \in \mathbb{R}_+$ determinat a.î. $t(\alpha) = f(\mathbf{x}_{k+1})$ să fie minimă[3].
Obs

1. dacă $\alpha_k \mathbf{S}_k = \mathbf{H}_k^{-1}$ - metoda Newton
2. $\mathbf{S}_k = \mathbf{I}$ - metoda celei mai rapide coborâri

Strategia:

$$\mathbf{S}_k \approx \mathbf{H}_k^{-1}$$

Poate fi mai de succes decât Newton dacă nu există garanţia că în punctul de minim matricea hessian e pozitiv definită.

[3] Algoritmul dat de relaţia (55) este cunoscut şi sub numele de *metoda gradienţilor deviaţi*, deoarece vectorul direcţie se obţine printr-o transformare liniară a gradientului (prin înmulţirea lui cu matricea $\mathbf{S}_k$.

Construcţia inversei H. Corecţia de rangul unu.

Ideal: aproximarea $\mathbf{S}_k$ să conveargă către inversa matricei Hessian în punctul soluţie şi metoda să se comporte global ca metoda Newton.

$$f(\mathbf{x}) \approx f_k + (\mathbf{x} - \mathbf{x}_k)^T \mathbf{g}_k + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^T \mathbf{H}_k (\mathbf{x} - \mathbf{x}_k)$$

$$\nabla f(\mathbf{x}) \approx \mathbf{g}_k + \mathbf{H}_k (\mathbf{x} - \mathbf{x}_k)$$

$\Rightarrow$ evaluat în $\mathbf{x}_{k+1}$

$$\mathbf{g}_{k+1} \approx \mathbf{g}_k + \mathbf{H}_k (\mathbf{x}_{k+1} - \mathbf{x}_k). \tag{56}$$

Notăm:

$$\mathbf{p}_k = \mathbf{x}_{k+1} - \mathbf{x}_k, \tag{57}$$

$$\mathbf{q}_k = \mathbf{g}_{k+1} - \mathbf{g}_k. \tag{58}$$

Construcţia inversei H. Corecţia de rangul unu.

$$\Rightarrow \quad \mathbf{q}_k = \mathbf{H}\mathbf{p}_k, \tag{59}$$

Evaluarea gradientului în două puncte dă informaţii despre matricea Hessian $\mathbf{H}$. Este natural să încercăm să construim aproximaţii succesive $\mathbf{S}_k$ ale inversei matricei Hessian bazate pe datele obţinute din primii k paşi ai procesului de coborâre astfel încât, dacă $\mathbf{H}$ ar fi constantă, aproximaţia să satisfacţa relaţia (59), adică:

$$\mathbf{S}_{k+1}\mathbf{q}_i = \mathbf{p}_i, \quad 0 \leq i \leq k. \tag{60}$$

După n paşi liniari independenţi se obţine $\mathbf{S}_n = \mathbf{H}^{-1}$. Deoarece $\mathbf{H}$ şi $\mathbf{H}^{-1}$ sunt simetrice, este natural să se construiască o aproximaţie $\mathbf{S}_k$ a lui $\mathbf{H}^{-1}$ care este de asemenea simetrică. Se caută o schemă care păstrează simetria:

$$\mathbf{S}_{k+1} = \mathbf{S}_k + \mathbf{z}_k\mathbf{z}_k^T, \tag{61}$$

Construcţia inversei H. Corecţia de rangul unu.

Vectorul coloană $\mathbf{z}_k$ defineşte o matrice care are rangul cel mult unu şi care corectează aproximarea inversei matricei Hessian. Le vom alege astfel încât relaţia (60) să fie satisfăcută. Luând *i* egal cu *k* în relaţia (60) şi folosind relaţia (61) se obţine:

$$\mathbf{p}_k = \mathbf{S}_{k+1}\mathbf{q}_k = \mathbf{S}_k\mathbf{q}_k + \mathbf{z}_k\mathbf{z}_k^T\mathbf{q}_k, \tag{62}$$

de unde rezultă că:

$$\mathbf{p}_k - \mathbf{S}_k\mathbf{q}_k = \mathbf{z}_k\mathbf{z}_k^T\mathbf{q}_k, \tag{63}$$

$$(\mathbf{p}_k - \mathbf{S}_k\mathbf{q}_k)^T = \mathbf{q}_k^T\mathbf{z}_k\mathbf{z}_k^T, \tag{64}$$

şi înmulţind aceste două relaţii rezultă că:

Construcţia inversei H. Corecţia de rangul unu.

$$\mathbf{z}_k \mathbf{z}_k^T = \frac{(\mathbf{p}_k - \mathbf{S}_k \mathbf{q}_k)(\mathbf{p}_k - \mathbf{S}_k \mathbf{q}_k)^T}{(\mathbf{z}_k^T \mathbf{q}_k)^2}. \tag{65}$$

şi, în final

$$\mathbf{S}_{k+1} = \mathbf{S}_k + \frac{(\mathbf{p}_k - \mathbf{S}_k \mathbf{q}_k)(\mathbf{p}_k - \mathbf{S}_k \mathbf{q}_k)^T}{\mathbf{q}_k^T (\mathbf{p}_k - \mathbf{S}_k \mathbf{q}_k)}. \tag{66}$$

Algoritmul metodei cvasi-Newton cu corecţie de rang 1

1. Alege $\mathbf{S}_0$ simetrică şi pozitiv definită
   Alege $\mathbf{x}_0$
   $k = 0$
   Calculează $\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$
2. **repetă**
   - 2.1. $\mathbf{v}_k = -\mathbf{S}_k \mathbf{g}_k$
   - 2.2. Minimizează $t(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{v}_k)$ în raport cu $\alpha \geq 0 \Rightarrow \alpha_k$
   - 2.3. $\mathbf{p}_k = \alpha_k \mathbf{v}_k$
   - 2.4. $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$
   - 2.5. Calculează $\mathbf{g}_{k+1} = \nabla f(\mathbf{x}_k)$
   - 2.6. $\mathbf{q}_k = \mathbf{g}_{k+1} - \mathbf{q}_k$
   - 2.7. $\mathbf{S}_{k+1} = \mathbf{S}_k + (\mathbf{p}_k - \mathbf{S}_k \mathbf{q}_k)(\mathbf{p}_k - \mathbf{S}_k \mathbf{q}_k)^T / (\mathbf{q}_k^T (\mathbf{p}_k - \mathbf{S}_k \mathbf{q}_k))$
   - 2.8 $k = k + 1$

   până când este îndeplinită condiţia de oprire

Algoritmul metodei cvasi-Newton cu corecţie de rang 1

Dificultăţi:

- $\mathbf{S}_{k+1}$ calculată cu formula (66) rămâne pozitiv definită numai dacă $\mathbf{q}_k^T(\mathbf{p}_k - \mathbf{S}_k\mathbf{q}_k) > 0$, condiţie care nu poate fi garantată.
- Chiar dacă$\mathbf{S}_{k+1}$ este pozitivă, în cazul în care ea este mică apar dificultăţi numerice.

Metoda Davidon-Fletcher-Powell

- Are o proprietate uimitoare: pentru o funcţie obiectiv pătratică generează simultan direcţiile din metoda gradienţilor conjugaţi şi construieşte inversa matricei Hessian.
- La fiecare pas aproximaţia inversei matricei Hessian este corectată prin intermediul a două matrice simetrice de rangul unu, şi de aceea această schemă este adesea numită procedura de corecţie de rangul doi.

Formula propusă pentru calculul aproximării inversei matricei Hessian:

$$\mathbf{S}_{k+1} = \mathbf{S}_k + \frac{\mathbf{p}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{q}_k} - \frac{\mathbf{S}_k \mathbf{q}_k \mathbf{q}_k^T \mathbf{S}_k}{\mathbf{q}_k^T \mathbf{S}_k \mathbf{q}_k}. \quad (67)$$

Algoritmul metodei Davidon-Fletcher-Powell

1. Alege $\mathbf{S}_0$ simetrică şi pozitiv definită
   Alege $\mathbf{x}_0$
   $k = 0$
   Calculează $\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$
2. **repetă**
   - 2.1. $\mathbf{v}_k = -\mathbf{S}_k \mathbf{g}_k$
   - 2.2. Minimizează $t(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{v}_k)$ în raport cu $\alpha \geq 0 \Rightarrow \alpha_k$
   - 2.3. $\mathbf{p}_k = \alpha_k \mathbf{v}_k$
   - 2.4. $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$
   - 2.5. Calculează $\mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1})$
   - 2.6. $\mathbf{q}_k = \mathbf{g}_{k+1} - \mathbf{q}_k$
   - 2.7. $\mathbf{S}_{k+1} = \mathbf{S}_k + \mathbf{p}_k \mathbf{p}_k^T / (\mathbf{p}_k^T \mathbf{q}_k) - \mathbf{S}_k \mathbf{q}_k \mathbf{q}_k^T \mathbf{S}_k / (\mathbf{q}_k^T \mathbf{S}_k \mathbf{q}_k)$
   - 2.8 $k = k + 1$

   până când este îndeplinită condiţia de oprire

Metoda Davidon-Fletcher-Powell

Se poate demonstra că

- Dacă $\mathbf{S}_k$ este pozitiv definită, atunci $\mathbf{S}_{k+1}$ este şi ea pozitiv definită. Este interesant că această afirmaţie este adevărată chiar dacă α_k nu este un punct de minim pentru funcţia $t(\alpha)$.
- Dacă f este o funcţie pătratică, având deci o matrice Hessian constantă $\mathbf{H}$, atunci metoda Davidon-Fletcher-Powell generează direcţii $\mathbf{p}_k$ care sunt $\mathbf{H}$-ortogonale, iar după n paşi $\mathbf{S}_n = \mathbf{H}^{-1}$. În acest caz, metoda face minimizări liniare succesive de-a lungul unor direcţii conjugate. Mai mult, dacă aproximarea iniţială $\mathbf{S}_0$ este luată matricea unitate, atunci metoda devine metoda gradienţilor conjugaţi iar soluţia se obţine după n paşi.

Metoda Broyden-Fletcher-Goldfarb-Shanno

La DFP, formula pentru calculul matricei $\mathbf{S}_{k+1}$ se bazează pe

$$\mathbf{S}_{k+1}\mathbf{q}_i = \mathbf{p}_i, \quad 0 \le i \le k, \tag{68}$$

care a fost dedusă din

$$\mathbf{q}_i = \mathbf{H}\mathbf{p}_i, \quad 0 \le i \le k, \tag{69}$$

relaţie care este adevărată dacă funcţia f este pătratică.
Altă idee: a folosi aproximaţii chiar ale matricei Hessian şi nu ale inversei ei.

- Notăm aproximaţiile lui $\mathbf{H}$ cu $\mathbf{T}_k$
- Vom căuta în mod analog să avem satisfăcute relaţiile:

$$\mathbf{q}_i = \mathbf{T}_{k+1}\mathbf{p}_i, \quad 0 \le i \le k. \tag{70}$$

Metoda Broyden-Fletcher-Goldfarb-Shanno

Se obţine:
Corecţia de rangul unu:

$$\mathbf{T}_{k+1} = \mathbf{T}_k + \frac{(\mathbf{q}_k - \mathbf{T}_k\mathbf{p}_k)(\mathbf{q}_k - \mathbf{T}_k\mathbf{p}_k)^T}{\mathbf{p}_k^T(\mathbf{q}_k - \mathbf{T}_k\mathbf{p}_k)}. \tag{71}$$

Şi corecţia de rangul doi (BFGS)

$$\mathbf{T}_{k+1} = \mathbf{T}_k + \frac{\mathbf{q}_k\mathbf{q}_k^T}{\mathbf{q}_k^T\mathbf{p}_k} - \frac{\mathbf{T}_k\mathbf{p}_k\mathbf{p}_k^T\mathbf{T}_k}{\mathbf{p}_k^T\mathbf{T}_k\mathbf{p}_k} \tag{72}$$

de unde

$$\mathbf{S}_{k+1}^{\mathrm{BFGS}} = \mathbf{S}_k + \left(\frac{1 + \mathbf{q}_k^T\mathbf{S}_k\mathbf{q}_k}{\mathbf{q}_k^T\mathbf{p}_k}\right)\frac{\mathbf{p}_k\mathbf{p}_k^T}{\mathbf{p}_k^T\mathbf{q}_k} - \frac{\mathbf{p}_k\mathbf{q}_k^T\mathbf{S}_k + \mathbf{S}_k\mathbf{q}_k\mathbf{p}_k^T}{\mathbf{q}_k^T\mathbf{p}_k} \tag{73}$$

În algoritm DFP, se schimbă pasul 2.7

DFP vs BFGS

- Experimentele numerice au arătat că performanţa formulei BFGS este superioară celei DFP, şi de aceea ea este preferată.
- Atât DFP cât şi BFGS folosesc o corecţie de rangul doi care este construită cu ajutorul vectorilor $\mathbf{p}_k$ şi $\mathbf{S}_k\mathbf{q}_k$. Combinaţii ponderate ale acestor formule vor fi de aceea de acelaşi tip (simetrice, de rangul doi, şi construite din $\mathbf{p}_k$ şi $\mathbf{S}_k\mathbf{q}_k$).

  Această observaţie a condus în mod natural la considerarea unei familii întregi de metode, cunoscute sub numele de metode de tip Broyden, definite de relaţia

$$\mathbf{S}^{\Phi} = (1 - \Phi)\mathbf{S}^{\mathrm{DFP}} + \Phi\mathbf{S}^{\mathrm{BFGS}}, \tag{74}$$

  unde Φ este un parametru care poate lua orice valoare reală.

Metrică variabilă sau gradienţi conjugaţi?

- GC este un caz particular al metodelor de metrică variabilă (MV).
- Se foloseşte informaţia obţinută din minimizări unidimensionale de-a lungul unor direcţii succesive.
- Algoritmii sunt construiţi astfel încât n minimizări liniare să conducă către minimul exact al unei funcţii pătratice în n dimensiuni.
- Pentru funcţii mai generale, nepătratice, direcţiile sunt întotdeauna coborâtoare, iar după n iteraţii convergenţa este superliniară.
- MV diferă de GC prin faptul că memorează şi reactualizează informaţia acumulată.

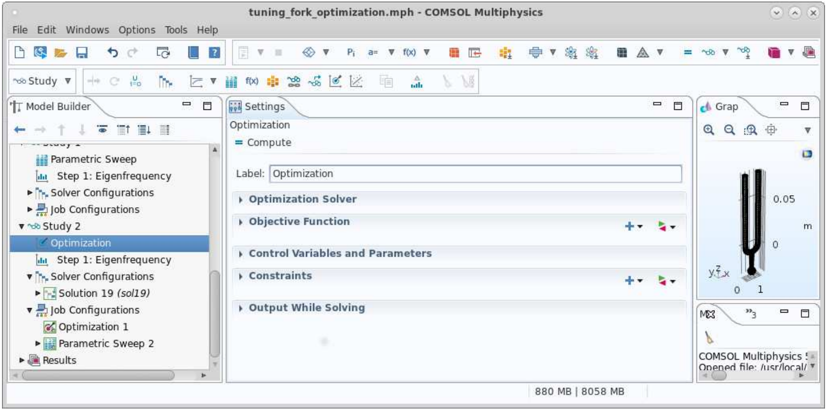
  În loc să memoreze un vector intermediar de dimensiune n, ele memorează o matrice de dimensiune $n \times n$. În general, pentru n moderat, acesta nu este un dezavantaj semnificativ.
- Există multe implementări sofisticate ale metodelor de MV, care minimizează eroarea de rotunjire sau tratează condiţii mai speciale.

Aspecte avansate

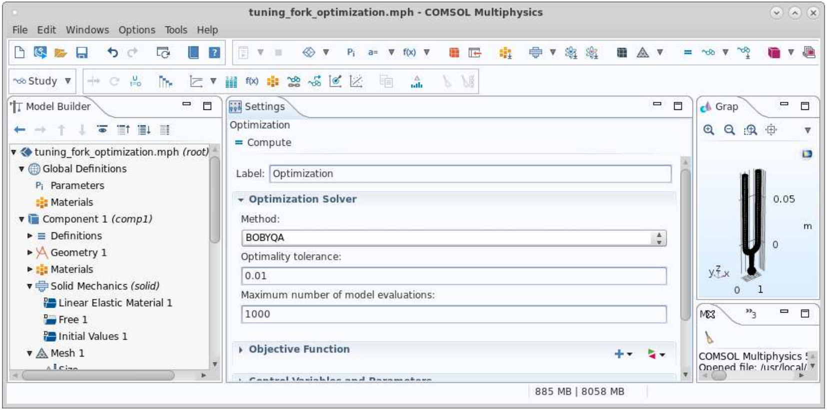
- Metode de tip *trust region* (de exemplu Levenberg-Marquardt, etc)

  [Yuan15] Ya-xiang Yuan, Recent advances in trust region algorithms, Mathematical Programming 151(1), 2015, disponibila la https://www.researchgate.net/publication/273908953_Recent_advances_in_trust_region_algorithms
- Tratarea restricţiilor
  - SQP
  - Interior point methods
  - Method of moving aysmptotes
  - etc

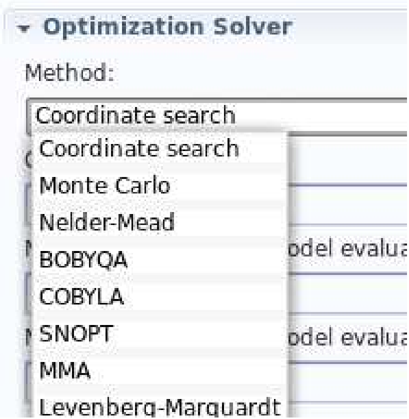
COMSOL - formularea prolemei de optimizare



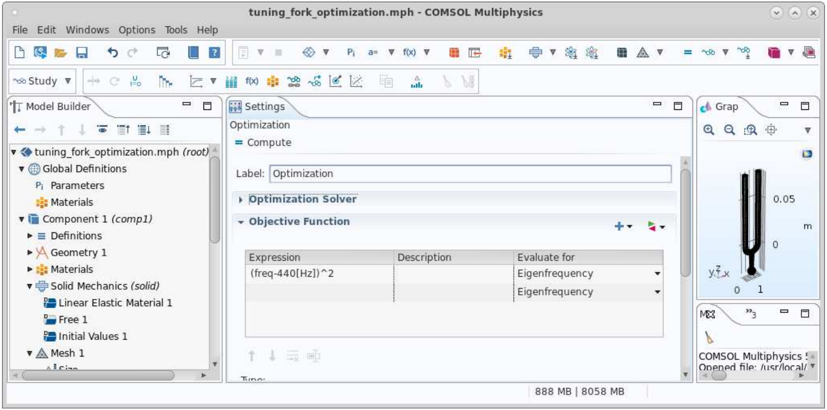
COMSOL - metode disponibile



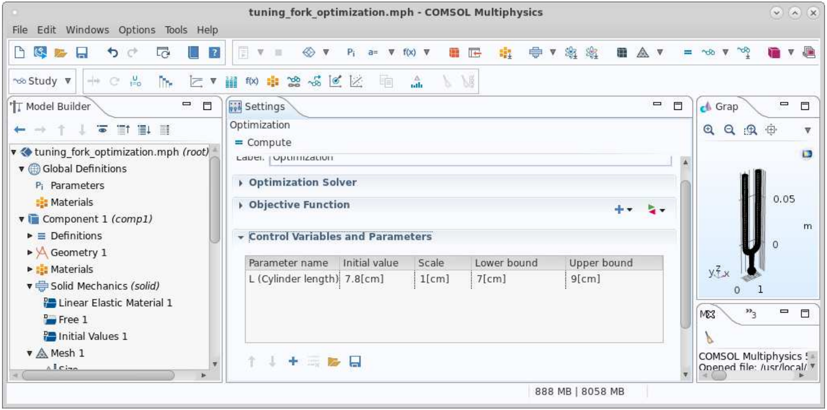
COMSOL - metode disponibile



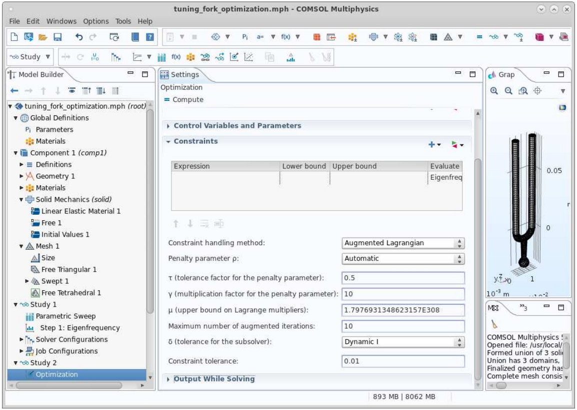
COMSOL - descrierea funcţiei obiectiv



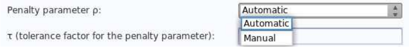
COMSOL - parametrii de optimizat şi domeniile lor

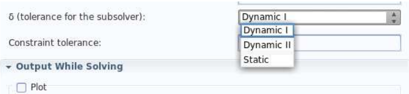


COMSOL - alte restricţii (info)



COMSOL - alte restricţii (alegeri posibile)





Optimization toolbox

Fără restricţii:

- Quasi-Newton
- Nelder-Mead:
- Regiunea de încredere (*trust region*) - utilă pentru probleme cu multe variabile, unde poate fi exploatată structura şi raritatea.

Cu restricţii

- Metode de punct interior;
- Programare pătratică secvenţială (SQP);

Alte metode şi detalii la https://ch.mathworks.com/products/optimization.html

Referinţe

- [Ciuprina02] G.Ciuprina, D.Ioan, I.Munteanu, M.Rebican, R.Popa, Optimizarea numerica a dispozitivelor electromagnetice, Editura Printech, 2002.

  disponibilă la http://www.lmn.pub.ro/∼gabriela/books/opt2002.pdf

- [Cheney08] Ward Cheney and David Kincaid, *Numerical Mathematics and Computing*, Brooks/Cole publishing Company,2008. (Capitolul 16 - Minimization of functions)

- [Press02] W.H.Press, S.A.Teukolsky, W.T. Wetterling, B.P. Flannery, *Numerical Recipies in C*, 2002. (Capitolul 10)

  Disponibilă la https://www2.units.it/ipl/students_area/imm2/files/Numerical_Recipes.pdf