

# Metoda quasi-Newton diagonală bazată pe minimizarea funcției Byrd-Nocedal pentru optimizare fără restricții

Neculai ANDREI

Institutul Național de Cercetare-Dezvoltare în Informatică – ICI București  
B-dul Mareșal Alexandru Averescu, Nr. 8-10, 011455, București, România  
nandrei@ici.ro

**Rezumat:** În această lucrare prezentăm o nouă paradigmă de optimizare fără restricții care constă în următoarele ingrediente: 1) aproximarea matricei Hessian a funcției de minimizat cu o matrice diagonală, în care elementele diagonale sunt pozitive, 2) determinarea valorilor elementelor diagonale prin minimizarea funcției măsură a lui Byrd și Nocedal referitor la ecuația secantei slabă a lui Dennis și Wolkowicz. Algoritmul obținut este foarte simplu, și pentru acesta se demonstrează convergența liniară. Performanțele numerice ale algoritmului sunt ilustrate prin rezolvarea unui tren de 80 de probleme de test de optimizare fără restricții de diverse structuri și complexități, precum și prin rezolvarea a cinci aplicații de optimizare fără restricții cu 10000 sau 40000 de variabile. Comparații numerice intensive ale performanțelor acestui algoritm versus algoritmi pasul descendent, Cauchy cu scalare Oren-Luenberger, sau BFGS arată superioritatea abordării propuse în ceea ce privește numărul de iterații și timpul de calcul pentru obținerea unei soluții optime locale. Noutatea în această abordare este utilizarea funcției măsură, care în esența ei este un ingredient foarte puternic pentru demonstrarea proprietăților teoretice ale algoritmilor de optimizare fără restricții. Noi am arătat aici importanța ei și în ceea ce privește generarea de algoritmi eficienți de optimizare fără restricții.

**Cuvinte cheie:** Optimizare fără restricții, Ecuația secantei slabă, Actualizare diagonală quasi-Newton, Funcția măsură Byrd-Nocedal, Comparații numerice.

## Diagonal quasi-Newton method based on minimizing the Byrd-Nocedal function for unconstrained optimization

**Abstract:** A new quasi-Newton method with a diagonal updating matrix is suggested, where the diagonal elements are determined by minimizing the measure function of Byrd and Nocedal subject to the weak secant equation of Dennis and Wolkowicz. The Lagrange multiplier of this minimization problem is computed by using an adaptive procedure based on the conjugacy condition. The convergence of the algorithm is proved for twice differentiable, convex and bounded below functions using only the trace and the determinant. Using a set of 80 unconstrained optimization test problems and some applications from the MINPACK-2 collection we have the computational evidence that the algorithm is more efficient and more robust than the steepest descent, than the Barzilai and Borwein algorithm, than the Cauchy algorithm with Oren and Luenberger scaling and than the classical BFGS algorithms with the Wolfe line search conditions.

**Keywords:** Unconstrained optimization. Weak secant. Diagonal quasi-Newton update. Measure function of Byrd and Nocedal. Numerical comparisons.

### 1. Introducere

Pentru rezolvarea problemei de optimizare fără restricții

$$\min f(x), \quad (1)$$

unde  $f: R^n \rightarrow R$  este o funcție continuu diferențiabilă mărginită inferior și  $x \in R^n$ , una dintre cele mai importante metode este metoda quasi-Newton. Plecând de la un punct inițial care aproximează soluția optimă a problemei (1)  $x_0 \in R^n$  și dispunând de o aproximație inițială  $B_0 \in R^{n \times n}$  a

Hessianului funcției  $f$ , simetrică și pozitiv definită, această metodă generează un șir de puncte, calculate prin formula iterativă:

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, \dots \quad (2)$$

unde direcția de căutare  $d_k \in R^n$  este aleasă ca soluție a sistemului algebric liniar:

$$B_k d_k = -g_k, \quad (3)$$

iar  $\alpha_k$  este lungimea pasului de deplasare de-a lungul acestei direcții. În (3),  $B_k \in R^{n \times n}$  este o aproximare a Hessianului  $\nabla^2 f(x_k)$  funcției  $f$  în punctul  $x_k$  și  $g_k$  este gradientul  $\nabla f(x_k)$  lui  $f$  în  $x_k$ .

Prima metodă quasi-Newton a fost propusă de Davidon [19], mai departe fiind analizată și dezvoltată de Fletcher și Powell [25]. Această metodă este cunoscută ca metoda DFP. Multe alte metode quasi-Newton au fost propuse în literatură, dar de-a lungul anilor după numeroase experimente numerice metoda BFGS [16, 26, 29, 43] s-a impus, fiind considerată ca și cea mai eficientă și robustă pentru optimizare fără restricții. În această metodă, aproximația Hessianului este calculată sub forma:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}, \quad (4)$$

$k = 0, 1, \dots$ , unde  $s_k = x_{k+1} - x_k$  și  $y_k = g_{k+1} - g_k$ . Deseori, în implementările practice, în locul formulei (3) direcția de căutare este calculată ca  $d_{k+1} = -H_{k+1} g_{k+1}$ , unde  $H_{k+1} = B_{k+1}^{-1}$ , este inversa aproximației BFGS a Hessianului. Utilizând de două ori formula Sherman-Morrison-Woodbury de rang unu, din (4) obținem:

$$H_{k+1} = H_k - \frac{s_k y_k^T H_k + H_k y_k s_k^T}{s_k^T y_k} + \left( 1 + \frac{y_k^T H_k y_k}{s_k^T y_k} \right) \frac{s_k s_k^T}{s_k^T y_k}. \quad (5)$$

Lungimea pasului de căutare  $\alpha_k$  se determină prin căutare liniară exactă:  $\min\{f(x_k + \alpha d_k), \alpha > 0\}$ , sau mai practic prin căutare liniară inexactă de tipul Wolfe [46, 47]:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \rho \alpha_k g(x_k)^T d_k, \quad (6)$$

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma g(x_k)^T d_k, \quad (7)$$

unde constantele pozitive  $\sigma$  și  $\rho$  satisfac condiția  $0 < \rho < \sigma < 1$ .

Importanța metodelor quasi-Newton a fost foarte repede recunoscută. Pentru a fi implementate, este necesară evaluarea gradientului funcției de minimizat. Totuși, unul dintre defectele majore constă în faptul că această metodă cere memorarea aproximației Hessianului sau a inversei acestuia. Deci, aceste metode sunt potrivite pentru rezolvarea problemelor de optimizare cu un număr mediu de variabile (câteva sute). Pentru a îmbunătăți performanțele metodelor quasi-Newton, mai multe modificări ale acestuia au fost propuse în literatură. De exemplu, anumite metode modifică calculul lungimii pasului, ca în căutarea liniară nemonotonă [32], altele utilizează o nouă căutare liniară de tip Armijo [44], sau o modificare a metodei BFGS nemonotone [45]. În acest articol vom dezvolta un alt mod de abordare bazat pe actualizarea diagonală a Hessianului utilizând minimizarea funcției măsură a lui Byrd și Nocedal.

Bertsekas [14, p. 67] a accentuat că pentru rezolvarea problemelor dificile, tendința este de a utiliza metode sofisticate ca metoda Newton sau metode de tip quasi-Newton. Deseori acesta este modul de abordare, mai ales pentru probleme prost condiționate. Totuși, metode mai simple cu

proceduri simple de calcul a direcției de căutare și a lungimii pasului de deplasare sunt mult mai profitabile decât metodele sofisticate. Aceasta este motivată și de faptul că proceduri sofisticate de calcul a direcției de căutare și reguli complicate de calcul a lungimii pasului sunt bazate pe presupuneri care nu sunt verificate în cazul problemelor dificile. Cu alte cuvinte, metode simple de optimizare sunt mult mai atractive și deseori mult mai eficiente pentru rezolvarea problemelor dificile. Totuși, trebuie să fie un echilibru, o balanță între simplu și simplist. Metoda prezentată în acest articol se încadrează în spiritual acestei observații.

O idee recentă de a genera algoritmi simpli de minimizare pentru optimizarea fără restricții constă în aproximarea Hessianului funcției de minimizat ca o matrice diagonală cu elemente pozitive pe diagonala principală. Acest mod de abordare a fost introdus de Nazareth [39], unde aproximarea diagonală a Hessianului este obținută utilizând ecuația secantei propusă de Dennis și Wolkowicz [21]. Zhu, Nazareth și Wolkowicz [48] au utilizat un principiu variațional pentru a determina elementele diagonale care satisfac: ecuația secantei, ecuația secantei modificată, metoda quasi-Cauchy, sau metoda quasi-Cauchy slabă. Variante ale acestui mod de abordare care nu implică căutarea liniară au fost date de Hassan, Leong și Farid [31] și de Leong, Hassan și Farid [33]. Alte metode îmbunătățite bazate pe metoda quasi-Cauchy modificată, care utilizează principiul variațional, au fost dezvoltate și analizate de Leong, Farid și Hassan [34] și Farid, Leong și Zheng [24]. Toate aceste metode pentru calculul unei actualizări diagonale a Hessianului sunt bazate pe tehnica variațională introdusă și utilizată pentru prima dată pentru a se obține actualizările quasi-Newton Powell-Symmetric-Broyden (PSB) sau Symmetric Rank One (SR1) (vezi Dennis și Schnabel [20]). La fel ca în metodele quasi-Newton PSB și SR1, defectul major constă în faptul că aceste actualizări bazate pe tehnici variaționale pot conduce la matrice care nu sunt pozitiv definite, astfel ruinând complet convergența metodei. Pentru a remedia acest defect, Leong, Farid și Hassan [35] au prezentat metode de scalare a actualizărilor diagonale quasi-Newton care conservă pozitiv definirea acestora.

În acest articol propunem o altă abordare în care Hessianul funcției de minimizat este aproximat ca o *matrice diagonală pozitiv definită*. Elementele acestei aproximări diagonale a Hessianului sunt determinate prin minimizarea *funcției măsură a lui Byrd și Nocedal* referitor la *ecuația secantei slabă*. Mai precis, elementele diagonale ale matricei care aproximează Hessianul funcției de minimizat sunt calculate prin minimizarea funcției măsură  $\varphi$  a lui Byrd și Nocedal [17] referitor la ecuația secantei slabă a lui Dennis și Wolkowicz [21]. Pentru a determina multiplicatorul Lagrange asociat acestei restricții se sugerează utilizarea condiției de conjugare într-o schemă adaptivă bazată pe polii ecuației secantă slabe.

Convergența globală se demonstrează pentru funcții convexe, mărginite inferior și diferențiabile, utilizând operatorii trace și determinant a matricei de iterație. Se demonstrează că atât pentru funcții pătratice cât și pentru cele nepătratice convergența este liniară. Experimente numerice intensive și comparații cu algoritmul lui Barzilai și Borwein [12], sau cu algoritmul Cauchy cu scalarea Oren și Luenberger în formă complementară [42] sau cu metoda BFGS clasică arată că metoda sugerată și algoritmul corespunzător este mult mai eficient și robust. Experimentele numerice utilizează 80 probleme de test de optimizare fără restricții descrise în [1], precum și aplicații ingineresti din colecția MINPACK-2 [8].

## 2. Calculul direcției de căutare

Byrd și Nocedal [17] au introdus funcția:

$$\varphi(A) = \text{tr}(A) - \ln(\det(A)), \quad (8)$$

definită pe mulțimea matricelor pozitiv definite, unde  $\ln(\cdot)$  reprezintă logaritmul natural. Fletcher [27] a observat că atât metoda BFGS cât și metoda DFP pot fi obținute printr-un argument variațional utilizând funcția  $\varphi$ . Observăm că funcția  $\varphi$  lucrează simultan cu trace (urma) și determinantul matricei, astfel simplificând analiza metodelor quasi-Newton. În fapt, această funcția este

o măsură a matricelor implicând toate valorile proprii ale lui  $A$ , și nu numai cea mai mică și cea mai mare, cum tradițional se utilizează în analiza metodelor quasi-Newton bazate pe numărul de condiționare (vezi [2, 3, 5, 10, 11]). Observăm că această funcție este strict convexă pe mulțimea matricelor simetrice și pozitiv definite, și este minimizată de  $A = I$ . Pe lângă acestea, ea este nemărginită când  $A$  devine singulară sau infinită, ceea ce înseamnă că aceasta lucrează ca o funcție barieră care păstrează  $A$  pozitiv definită.

În cele ce urmează vom defini algoritmul nostru prin minimizarea acestei funcții referitor la ecuația secantei slabă a lui Dennis și Wolkowicz. Considerăm:

$$B_{k+1} = \text{diag}(b_{k+1}^1, \dots, b_{k+1}^n) \quad (9)$$

unde  $b_{k+1}^i > 0$ , pentru toți  $i = 1, \dots, n$ , ca o matrice diagonală care multiplică gradientul cu semn schimbat în direcția de căutare, adică direcția de căutare este calculată ca  $d_{k+1} = -B_{k+1}^{-1}g_{k+1}$ , unde  $B_{k+1}$  este dat de (9). Deoarece  $B_{k+1}$  este o matrice diagonală și pozitiv definită, rezultă că în acest algoritm fiecare componentă a gradientului cu semn schimbat este înmulțită cu diferiți factori pozitivi. Notăm că această metodă cere numai  $O(n)$  locații de memorie pentru a memora  $B_{k+1}$ . În contrast, în metoda quasi-Newton,  $O(n^2)$  locații de memorie sunt necesare pentru a memora matricea  $n \times n$  dimensională care aproximează Hessianul.

După cum am zis, matricea  $B_{k+1}$  în (9) este determinată ca soluție a problemei:

$$\min \varphi(B_{k+1}) = \text{tr}(B_{k+1}) - \ln(\det(B_{k+1})) \quad (10)$$

referitor la .....

$$s_k^T B_{k+1} s_k = s_k^T y_k,$$

unde restricția din (10) este ecuația secantei slabă a lui Dennis și Wolkowicz [21]. Motivația utilizării ecuației secante slabă (10) este următoarea. Din ecuația secantei  $B_{k+1}s_k = y_k$  vedem că  $y$  este o aproximare a lui  $\nabla^2 f(x)s$ . Dar,  $y$  este un vector și teorema de medie poate să nu fie adevărată pentru funcții de mai multe variabile. Nu cunoaștem dacă sau nu există un vector  $\bar{x} \in \mathbb{R}^n$  astfel încât  $y = \nabla^2 f(\bar{x})s$  (vezi Dennis și Schnabel [20, pagina 74]). Totuși, teorema lui Taylor și teorema de medie ne asigură că există  $\bar{x}$  astfel încât  $s^T y = s^T \nabla^2 f(\bar{x})s$ . Deci, pare rezonabil să presupunem că matricea diagonală  $B_{k+1}$  satisface ecuația secantei slabă  $s_k^T B_{k+1} s_k = s_k^T y_k$  exprimată ca restricție în (10).

Acum, având în vedere că  $\text{tr}(B_{k+1}) = b_{k+1}^1 + \dots + b_{k+1}^n$  și  $\det(B_{k+1}) = b_{k+1}^1 \dots b_{k+1}^n$ , problema de minimizare (10) devine:

$$\min(b_{k+1}^1 + \dots + b_{k+1}^n - \ln(b_{k+1}^1 \dots b_{k+1}^n)) \quad (11)$$

referitor la: .....

$$\sum_{i=1}^n (s_k^i)^2 b_{k+1}^i = s_k^T y_k,$$

unde  $s_k^i$ ,  $i = 1, \dots, n$ , sunt componentele vectorului  $s_k$ . Lagrangianul problemei (11) este:

$$L = (b_{k+1}^1 + \dots + b_{k+1}^n) - \ln(b_{k+1}^1 \dots b_{k+1}^n) + \lambda \left( \sum_{i=1}^n (s_k^i)^2 b_{k+1}^i - s_k^T y_k \right), \quad (12)$$

unde  $\lambda$  este multiplicatorul Lagrange. Soluția problemei (11) este un punct staționar al Lagrangianului [4]. Deci, din (12) obținem:

$$\frac{\partial L}{\partial b_{k+1}^i} = 1 - \frac{1}{b_{k+1}^i} + \lambda (s_k^i)^2 = 0, \quad i = 1, \dots, n. \quad (13)$$

Din (13), elementele diagonale ale matricei  $B_{k+1}$  sunt determinate ca:

$$b_{k+1}^i = \frac{1}{1 + \lambda (s_k^i)^2}, \quad i = 1, \dots, n. \quad (14)$$

Deoarece  $d_{k+1} = -B_{k+1}^{-1} g_{k+1}$  din (9) și (14) rezultă că

$$d_{k+1}^i = -g_{k+1}^i (1 + \lambda (s_k^i)^2), \quad (15)$$

unde  $d_{k+1}^i$  și  $g_{k+1}^i$ ,  $i = 1, \dots, n$ , sunt respectiv componentele vectorilor  $d_{k+1}$  și  $g_{k+1}$ . Observăm că fiecare componentă a direcției de căutare  $d_{k+1}$  este dependentă de multiplicatorul Lagrange  $\lambda$ . Pentru a determina o valoare pentru  $\lambda$ , o metodă directă este de a rezolva sistemul neliniar dat de restricțiile din (10),  $F(\lambda) = 0$ , unde

$$F(\lambda) = \sum_{i=1}^n \frac{(s_k^i)^2}{1 + \lambda (s_k^i)^2} - s_k^T y_k. \quad (16)$$

Vedem că această funcție este independentă de minimizarea funcției  $f$ . Observăm că  $F(\lambda)$  are ca poli valorile:  $-1 / (s_k^i)^2$ ,  $i = 1, \dots, n$ . Toți acești poli sunt negativi. Considerăm

$$r = -1 / (s_k^j)^2 = \max_{i=1, \dots, n; s_k^i \neq 0} \{-1 / (s_k^i)^2\}. \quad (17)$$

valuarea maximă a polilor. Derivata lui  $F(\lambda)$  este:

$$F'(\lambda) = - \sum_{i=1}^n \frac{1}{(\lambda + 1 / (s_k^i)^2)^2} \quad (18)$$

și observăm că  $F'(\lambda) < 0$  pe intervalul

$$\left( -\frac{1}{(s_k^j)^2}, +\infty \right). \quad (19)$$

Deci,  $F(\lambda)$  este strict descrescătoare în intervalul de mai sus de la  $+\infty$  la  $-s_k^T y_k$ . Din condițiile de căutare liniară Wolfe,  $s_k^T y_k > 0$ . Deci, există o soluție unică  $\lambda^*$  în intervalul (19) astfel încât  $F(\lambda^*) = 0$ . Funcția  $F(\lambda)$  este destul de complexă pe domeniul ei de definiție și are  $n$  zerouri. Pentru a menține pozitivă definiția aproximației diagonale a Hessianului  $B_{k+1}$  suntem interesați numai de zeroul cel mai mare. Observăm că cu această soluție din intervalul (19) elementele diagonale  $b_{k+1}^i$ ,  $i = 1, \dots, n$ , ale matricei  $B_{k+1}$  sunt toate pozitive.

Totuși, această metodă bazată pe rezolvarea ecuației neliniare  $F(\lambda) = 0$ , astfel încât  $b_{k+1}^i > 0$ ,  $i = 1, \dots, n$ , este prea complicată pentru a fi implementată. Pentru obținerea unei valori a lui  $\lambda$ , presupunem că direcția de căutare satisface condiția de conjugare:

$$y_k^T d_{k+1} = -t s_k^T g_{k+1}, \quad (20)$$

unde  $t$  este un parametru care va fi discutat mai târziu. Introducând (15) în (20), se obține următoarea valoare pentru  $\lambda$  :

$$\bar{\lambda} = \frac{t(s_k^T g_{k+1}) - y_k^T g_{k+1}}{\sum_{i=1}^n (y_k^i g_{k+1}^i (s_k^i)^2)}, \quad (21)$$

unde  $y_k^i$ ,  $i=1, \dots, n$ , sunt componentele vectorului  $y_k$ . Întrucât  $B_{k+1}$  din (9) trebuie să fie pozitiv definită, adică din (14)  $b_{k+1}^i > 0$ , pentru toți  $i=1, \dots, n$ , sugerăm calculul multiplicatorului Lagrange într-o manieră multiplicativă ca:

$$\lambda = \begin{cases} r + \theta, & \text{if } \bar{\lambda} < r, \\ \bar{\lambda} & \text{if } \bar{\lambda} \geq r, \end{cases} \quad (22)$$

unde  $r$  este polul maxim al lui  $F(\lambda)$  și  $\theta$  este o mică perturbare a valorii maxime a polului, introdusă pentru a evita singularitățile (de exemplu  $\theta = 1$ ). Motivația utilizării strategiei adaptive de mai sus pentru calculul lui  $\lambda$  dată de (22) este că suntem interesați de o soluție aproximativă a ecuației  $F(\lambda) = 0$  în intervalul  $(r, +\infty)$ . Desigur că orice  $\lambda \in (r, +\infty)$  determină  $b_{k+1}^i > 0$ , pentru toți  $i=1, \dots, n$ . Totuși preferăm un punct care satisface condiția de conjugare.

### Actualizare diagonală utilizând minimizarea funcției $\varphi$ - MINFI

1.	Inițializare. Se alege un punct inițial $x_0 \in R^n$ , constantele $\sigma$ , $\rho$ cu $0 < \rho < \sigma < 1$ , o valoare pozitivă a parameterului $t > 0$ , o valoare pozitivă a parameterului $\theta$ și $\varepsilon > 0$ suficient de mic. Se calculează $g_0 = \nabla f(x_0)$ . Se pune $d_0 = -g_0$ și $k = 0$
2.	Se testează un criteriu de oprire a iterațiilor. De exemplu, dacă $\ g_k\  \leq \varepsilon$ , atunci stop. Altfel, se continuă cu pasul 3
3.	Se determină lungimea pasului $\alpha_k > 0$ care satisface condițiile Wolfe (6) și (7)
4.	Se calculează $x_{k+1} = x_k + \alpha_k d_k$ , $f_{k+1} = f(x_{k+1})$ și $g_{k+1} = \nabla f(x_{k+1})$ . Se pune $s_k = x_{k+1} - x_k$ , $y_k = g_{k+1} - g_k$
5.	Se calculează multiplicatorul Lagrange $\lambda$ din (22), unde $\bar{\lambda}$ este calculat ca în (21) și $r$ este determinat ca în (17)
6.	Se calculează direcția de căutare $d_{k+1}$ , unde componentele ei sunt calculate ca în (15)
7.	Se pune $k = k + 1$ și se continuă cu pasul 2. ■

Observăm că direcția de căutare este calculată ca  $d_{k+1} = -B_{k+1}^{-1} g_{k+1}$ , unde elementele diagonale ale matricii  $B_{k+1}$  sunt determinate ca în (14). Deoarece  $b_{k+1}^i > 0$  pentru toți  $i=1, \dots, n$ , rezultă că

$$g_{k+1}^T d_{k+1} = -g_{k+1}^T B_{k+1}^{-1} g_{k+1} = -\sum_{i=1}^n (g_{k+1}^i)^2 (1 + \lambda (s_k^i)^2) < 0,$$

adică direcția  $d_{k+1}$  cu componentele date de (15) este o direcție descendentă. Suplimentar, vedem că pentru orice  $i=1, \dots, n$ ,  $s_k^i \rightarrow 0$  când  $k \rightarrow \infty$ . Din (15) observăm că direcția de căutare este o perturbare aditivă a direcției pasului descendent.

### 3. Convergența globală

În cele ce urmează demonstrăm că algoritmul MINFI cu căutare liniară inexactă este global convergent pentru minimizarea funcțiilor convexe.

Din (14) vedem că

$$\text{tr}(B_{k+1}) = \sum_{i=1}^n \frac{1}{1 + \lambda(s_k^i)^2} \quad \text{și} \quad \det(B_{k+1}) = \prod_{i=1}^n \left( \frac{1}{1 + \lambda(s_k^i)^2} \right). \quad (23)$$

Deoarece din (22)  $\lambda > 0$ , rezultă că  $1/(1 + \lambda(s_k^i)^2) \leq 1$ . Deci,

$$\text{tr}(B_{k+1}) \leq n \quad \text{și} \quad \det(B_{k+1}) \leq 1. \quad (24)$$

**Teorema 3.1:** *Presupunem că funcția  $f$  este convexă, mărginită inferior și de două ori continuu diferențiabilă în  $R^n$ . Atunci algoritmul MINFI este global convergent, adică*

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0. \quad (25)$$

**Demonstrație:**

Din (6) obținem:

$$f_{k+1} - f_0 \leq \sigma \sum_{i=0}^k s_i^T g_i.$$

Deoarece funcția  $f$  este mărginită inferior, rezultă că

$$-\sum_{k=0}^{\infty} s_k^T g_k < \infty. \quad (26)$$

Acum, să presupunem prin contradicție că

$$\liminf_{k \rightarrow \infty} \|g_k\| \neq 0. \quad (27)$$

Atunci există o constantă pozitivă  $\Gamma > 0$  astfel încât pentru toți  $k$ :

$$\|g_k\| \geq \Gamma. \quad (28)$$

$s_k = -\alpha_k B_k^{-1} g_k$ , și  $\|B_k\| \leq \text{tr}(B_k)$ , atunci utilizând (28) obținem

$$-\sum_{k=0}^{\infty} s_k^T g_k = \sum_{k=0}^{\infty} \alpha_k g_k^T B_k^{-1} g_k \geq \Gamma^2 \sum_{k=0}^{\infty} \frac{\alpha_k}{\text{tr}(B_k)}.$$

Deci,

$$\sum_{k=0}^{\infty} \frac{\alpha_k}{\text{tr}(B_k)} < \infty. \quad (29)$$

Din (24) vedem că urma matricei  $B_k$  și determinatul ei sunt mărginite. Deci, ambele șiruri  $\{B_k\}$  și  $\{B_k^{-1}\}$  sunt uniform mărginite superior. Cu aceasta putem concludiona că

$$\cos \theta_k = \frac{s_k^T B_k s_k}{\|s_k\| \|B_k s_k\|},$$

unghiul dintre direcția pasului descendent dată de  $-g_k$  și direcția de căutare  $d_k$  dată de algoritmul MINFI este mărginit față de zero. Aceasta, împreună cu presupunerea (28) dă o contradicție, deoarece rezultatul lui Zoutendijk [49] (vezi de asemenea Wolfe [46, 47]) afirmă că orice metodă de tipul (2) - (3) satisface

$$\sum_{k=0}^{\infty} \|g_k\|^2 \cos^2 \theta_k < \infty.$$

Deci această contradicție arată că  $\liminf_{k \rightarrow \infty} \|g_k\| = 0$ . ■

În cele ce urmează vom investiga convergența algoritmului MINFI pentru rezolvarea problemelor pătratice cu căutare liniară exactă. Să considerăm minimizarea funcției pătratice:

$$f(x) = \frac{1}{2} x^T Q x - x^T b, \quad (30)$$

unde  $Q \in R^{n \times n}$  este o matrice simetrică și pozitiv definită. În acest caz algoritmul bazat pe minimizarea funcției măsură a lui Byrd și Nocedal este

$$x_{k+1} = x_k - \alpha_k B_k^{-1} g_k, \quad (31)$$

unde

$$\alpha_k = \frac{g_k^T B_k^{-1} g_k}{g_k^T B_k^{-1} Q B_k^{-1} g_k}, \quad (32)$$

$$g_k = Q x_k - b \quad (33)$$

și  $B_k$  este o matrice diagonală, pozitiv definită cu elementele diagonale date de (14).

**Teorema 3.2:** Fie  $x^*$  punctul de minim unic al funcției  $f$  definită de (30) și definim

$$E(x) = \frac{1}{2} (x - x^*)^T Q (x - x^*). \quad (34)$$

Atunci pentru algoritmul (31) la fiecare iterație  $k$  are loc:

$$E(x_{k+1}) \leq \left( \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \right)^2 E(x_k), \quad (35)$$

unde  $\lambda_1$  și  $\lambda_n$  sunt cea mai mică și respectiv cea mai mare valoare proprie a matricei  $B_k^{-1} Q$ .

**Demonstrație:**

Prin substituție directă obținem:

$$\frac{E(x_k) - E(x_{k+1})}{E(x_k)} = \frac{(g_k^T B_k^{-1} g_k)^2}{(g_k^T B_k^{-1} Q B_k^{-1} g_k)(g_k^T Q^{-1} d_k)}.$$

Notăm:

$$R_k = (B_k^{-1})^{1/2} Q (B_k^{-1})^{1/2} \quad \text{și} \quad p_k = (B_k^{-1})^{1/2} g_k.$$

Deci,

$$\frac{E(x_k) - E(x_{k+1})}{E(x_k)} = \frac{(p_k^T p_k)^2}{(p_k^T R_k p_k)(p_k^T R_k^{-1} p_k)}.$$

Acum, din inegalitatea lui Kantorovich (vezi [37, pagina 235]) obținem:

$$E(x_{k+1}) \leq \left( \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \right)^2 E(x_k), \quad (36)$$

unde  $\lambda_1$  și  $\lambda_n$  sunt cea mai mică, respectiv cea mai mare valoare proprie a lui  $R_k$ . Dar  $R_k$  este similară cu  $B_k^{-1} Q$ , adică au aceleași valori proprii. ■



Teorema zice că pentru programarea pătratică (30), dacă  $B_k^{-1}$  în (31) este cât mai apropiată posibil de  $Q^{-1}$ , atunci ambele valori proprii  $\lambda_1$  și  $\lambda_n$  sunt apropiate de unitate și deci convergența este rapidă. Dacă  $B_k^{-1}Q$  are o valoare proprie la mare distanță de celelalte, atunci convergența va fi lentă. În cazul nostru,  $B_k^{-1}$  din (31), generat de metoda de actualizare quasi-Newton diagonală cu minimizarea funcției măsură Byrd-Nocedal (8), este diagonală, pozitiv definită, departe de  $Q^{-1}$ .

Cu alte cuvinte, din (35) vedem că convergența algoritmului MINFI pentru programarea pătratică este numai liniară.

*Exemplu ilustrativ.* Considerăm, de exemplu, minimizarea funcției  $\frac{1}{2} \sum_{i=1}^n ix_i^2$ , unde  $n = 100$  și  $x_0 = [2, \dots, 2]$ . Soluția acestei probleme este  $x^* = [0, \dots, 0]$  și  $f(x^*) = 0$ . Metoda pasului descendent furnizează soluția optimă în 492 iterații și 527 evaluări ale gradientului. Pe de altă parte, MINFI dă aceeași soluție în 48 iterații și 92 evaluări ale gradientului. De-a lungul iterațiilor, pentru MINFI avem  $0.91612 \leq ((\lambda_n - \lambda_1) / (\lambda_n + \lambda_1))^2 \leq 0.99999$ , adică convergența este liniară. Diferența dintre algoritmi este mai dramatică pentru probleme de mari dimensiuni. De exemplu, pentru  $n = 1000$ , pasul descendent necesită 5026 iterații și 5070 evaluări ale gradientului. Pe de altă parte, MINFI necesită numai 183 iterații și 282 evaluări ale gradientului. ■

În general, pentru funcții nepătratice, în loc de  $Q$  avem Hessianul  $\nabla^2 f(x_k)$ . Dar, în acest caz, la fel,  $B_k$ , dat de (9) cu elementele calculate ca în (14), este o matrice diagonală, pozitiv definită care este departe de  $\nabla^2 f(x_k)^{-1}$ . Totuși, în acest caz linia  $i$ -th a Hessianului înmulțită cu  $1 + \lambda(s_k^i)^2$ , modifică valorile proprii ale lui  $B_k^{-1}\nabla^2 f(x_k)$  astfel încât  $((\lambda_n - \lambda_1) / (\lambda_n + \lambda_1))^2 < 1$ . Deci, pentru funcții obiectiv generale (nepătratice) convergența lui MINFI este de asemenea liniară.

#### 4. Rezultate numerice

În această secțiune prezentăm rezultate numerice cu o implementare Fortran a algoritmului MINFI. Algoritmul MINFI este comparat cu pasul descendent, cu algoritmul lui Barzilai și Borwein [12], cu algoritmul Cauchy cu scalarea Oren și Luenberger în formă complementară [42] și cu algoritmul BFGS clasic. Componentele direcției de căutare în MINFI sunt calculate ca în (15). Referitor la valoarea parameterului  $t$  din condiția de conjugare (20), în experimentele noastre am găsit că valoarea  $t = y_k^T s_k$  este una dintre cele mai bune. Direcția de căutare în algoritmul pasului descendent este calculată ca  $d_{k+1} = -g_{k+1}$ . Direcția de căutare în algoritmul BFGS este calculată sub forma  $d_{k+1} = -H_{k+1}g_{k+1}$ , unde  $H_{k+1}$  este dat în (5).

În toți algoritmi, cu excepția algoritmului lui Barzilai și Borwein, lungimea pasului este determinată prin condițiile de căutare liniară Wolfe date de (6) și (7), unde  $\sigma = 0.8$  și  $\rho = 0.0001$ . Iterațiile sunt oprite dacă inegalitatea  $\|g_k\|_\infty \leq \varepsilon_g$  este satisfăcută, unde  $\varepsilon_g$  este o valoare pozitivă suficient de mică, și  $\|\cdot\|_\infty$  este componenta cu valoarea absolută maximă a unui vector, sau dacă numărul de iterații depășește o valoare întreagă prespecificată. Toate programele sunt scrise în Fortran, dublă precizie și compilate cu f77 (default compiling settings) pe o mașină Intel Pentium 4, 1.8GHz.

În acest studiu numeric am considerat un număr de 80 de probleme de test de optimizare fără restricții cu numărul de variabile limitat la 10000, prezentate în [1]. Algoritmi pe care-i comparăm furnizează soluții optime locale. Deci, comparația între aceștia se face în următorul context. Fie  $f_i^{ALG1}$  și  $f_i^{ALG2}$  valorile optime găsite de ALG1 și respective de ALG2 pentru problema

$i = 1, \dots, 80$ . Zicem că, pentru problema particulară  $i$ , performanța lui ALG1 a fost mai bună decât a lui ALG2 dacă:

$$|f_i^{ALG1} - f_i^{ALG2}| < 10^{-3} \quad (37)$$

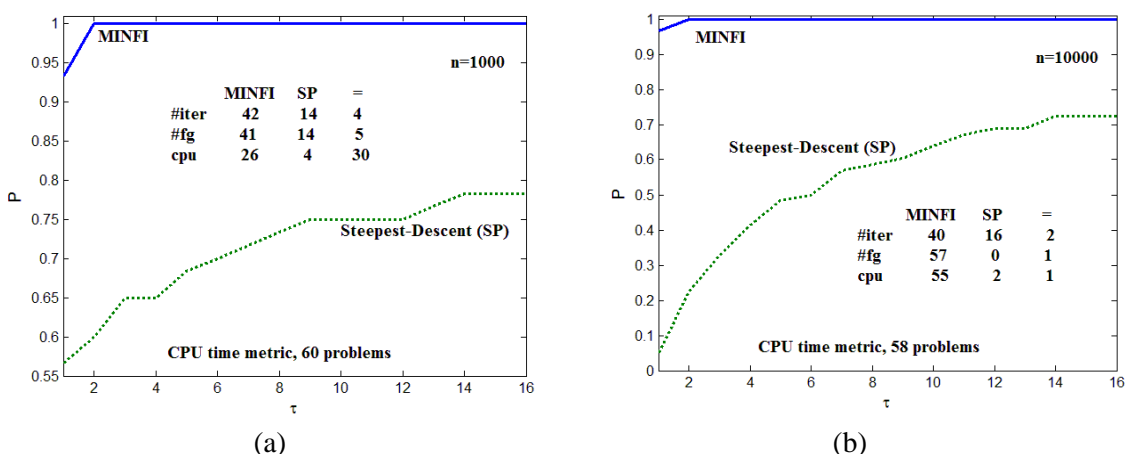
și dacă numărul de iterații (#iter), sau numărul de evaluări ale funcției și gradientului (#fg), sau timpul de calcul CPU al lui ALG1 a fost mai mic decât numărul de iterații, sau decât numărul de evaluări ale funcției și gradientului, sau decât timpul de calcul CPU corespunzător lui ALG2.

În primul set de experimente numerice comparăm MINFI versus pasul descendent (SP). Tabelul 1 prezintă performanțele lui MINFI versus SP pentru diferite valori ale lui  $\varepsilon_g$  pentru rezolvarea setului de 80 de probleme, fiecare dintre acestea cu 1000 variabile. Clar, ambii algoritmi au o rată de convergență liniară. Referitor la timpul de calcul CPU, pentru toate valorile lui  $\varepsilon_g$ , algoritmul MINFI este cel mai bun în comparație cu pasul descendent. Pentru a obține o soluție, SP cere mai multe iterații decât MINFI. Referitor la numărul de evaluări ale funcțiilor de minimizat și a gradientului acestora, vedem că MINFI este cel mai bun. Chiar dacă multiplicatorul Lagrange  $\lambda$  nu este calculat ca soluție a ecuației  $F(\lambda) = 0$  în intervalul  $(r, +\infty)$ , valoarea obținută prin schema adaptivă (22) este acceptabilă pentru performanțele lui MINFI.

**Tabelul 1.** Performanțele lui MINFI versus SP.  
 $n = 1000$ . maxiter = 50000.

$\varepsilon_g$	MINFI			SP		
	#iter	#fg	cpu	#iter	#fg	cpu
$10^{-4}$	344752	442259	64.09	932234	1121312	202.79
$10^{-5}$	469737	584687	84.65	1237036	1511493	254.70
$10^{-6}$	598286	775514	222.95	1377944	1772669	398.86

Figurile 1(a) și 1(b) prezintă profilul de performanță Dolan și Moré [22] asociat acestor algoritmi, pentru  $n = 1000$  și respectiv pentru  $n = 10000$ , referitor la timpul de calcul CPU, unde  $\varepsilon_g = 10^{-4}$  și numărul de iterații este limitat la 50000.



**Figura 1.** MINFI versus SP. (a)  $n = 1000$ . (b)  $n = 10000$ .

Din Figura 1(a), când comparăm MINFI versus SP, referitor la numărul de iterații, vedem că MINFI a fost mai bun în 42 probleme (adică a realizat numărul minim de iterații în rezolvarea a 42 de probleme), în timp ce SP a fost mai bun în 14 probleme. Ambii algoritmi au obținut același

număr de iterații în rezolvarea numai a 4 probleme, etc. Din 80 de probleme considerate în acest studiu numeric, numai pentru 60 dintre ele criteriul (37) a fost satisfăcut.

Din profilele de performanță arătate în Figurile 1(a) și 1(b) vedem că MINFI este în top față de SP, și diferențele sunt semnificative. Deoarece ambele programe, care implementează acești algoritmi, utilizează aceleași condiții Wolfe pentru determinarea lungimii pasului și același criteriu de oprire a iterațiilor, vedem că acestea diferă numai în procedura de calcul a direcției de deplasare. Procentul de probleme pentru care un algoritm este mai rapid este dat în partea din stânga a figurii. Partea din dreapta dă procentul de probleme care au fost rezolvate cu succes de acești algoritmi. Cu alte cuvinte, partea din stânga figurii arată eficiența unui algoritm, partea din dreapta arată robustețea lui.

În al doilea set de experimente numerice comparăm MINFI versus algoritmul lui Barzilai și Borwein (BB) [12]. În BB aproximarea soluției problemei (1) este calculată ca:

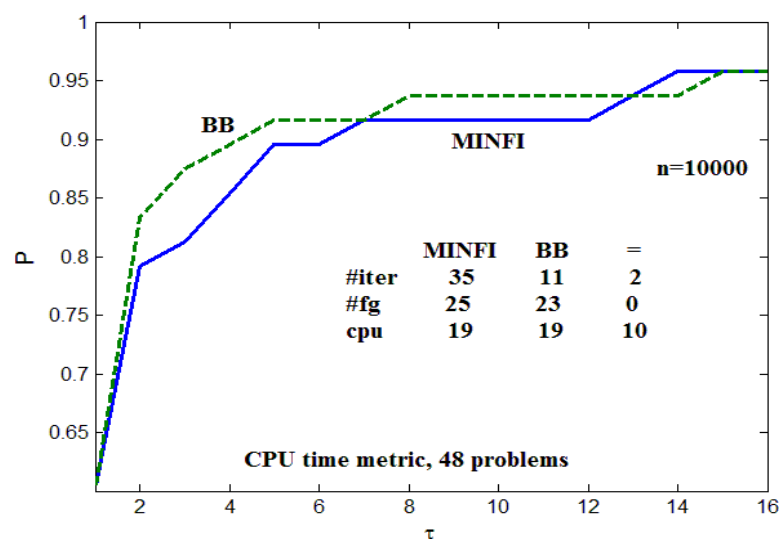
$$x_{k+1} = x_k - \frac{y_k^T s_k}{\|y_k\|^2} g_k. \quad (38)$$

Tabelul 2 prezintă performanțele acestor algoritmi pentru  $n=1000$  și diferite valori ale lui  $\varepsilon_g$ . Numărul maxim de iterații este limitat la 50000.

**Tabelul 2.** Performanțele lui MINFI versus BB.  
 $n = 1000$ . maxiter = 50000.

$\varepsilon_g$	MINFI			BB		
	#iter	#fg	cpu	#iter	#fg	cpu
$10^{-4}$	344752	442259	64.09	278811	278886	53.94
$10^{-5}$	469737	584687	84.65	376163	376237	105.52
$10^{-6}$	598286	775514	222.95	392792	292865	107.53

Figura 2 prezintă profilul de performanță Dolan și Moré al acestor algoritmi referitor la timpul de calcul CPU, unde numărul de variabile este 10000,  $\varepsilon_g = 10^{-4}$  și numărul maxim de iterații este limitat la 100000.



**Figura 2.** MINFI versus BB.  $n = 10000$ .

Din Tabelul 2 vedem că BB este ușor mai rapid, necesitând mai puține iterații și un număr mai mic de evaluări ale gradientului. Referitor la timpul de calcul CPU, Figura 2 arată că ambii algoritmi au aceeași eficiență și aceeași robustețe, BB fiind ușor mai rapid. MINFI consumă mult timp pentru calculul lungimii pasului  $\alpha_k$  utilizând condițiile Wolfe (6) și (7).

Pe de altă parte, BB fiind un algoritm de pas descendent în două puncte este foarte rapid, necesitând numai două produse scalare pe iterație.

În al treilea set de experimente numerice comparăm MINFI versus algoritmul lui Cauchy cu scalarea Oren și Luenberger [42], unde de data aceasta numărul de variabile este limitat la 100. În algoritmul lui Cauchy cu scalarea Oren și Luenberger în formă complementară (COL), iterațiile sunt calculate ca:

$$x_{k+1} = x_k - \alpha_k \frac{y_k^T s_k}{\|y_k\|^2} g_k, \quad (39)$$

unde  $\alpha_k$  este determinat de condițiile Wolfe (6) și (7). Prima iterație a acestui algoritm este dată de pasul descendent [48, 42].

Tabelul 3 prezintă performanțele acestor algoritmi pentru diferite valori ale lui  $\varepsilon_g$ , numărul maxim de iterații fiind limitat la 50000.

**Tabelul 3.** Performanțele lui MINFI versus COL.  
 $n = 100$ . maxiter = 50000.

$\varepsilon_g$	MINFI			COL		
	#iter	#fg	cpu	#iter	#fg	cpu
$10^{-4}$	201082	232453	4.20	613693	686025	49.97
$10^{-5}$	230186	270297	4.77	805629	882241	55.43
$10^{-6}$	317733	408344	6.76	909057	1000294	79.32

Observăm diferențele dintre acești algoritmi. În algoritmul COL, la iterația  $k$  toate componentele cu semn schimbat ale gradientului sunt scalate cu același factor  $y_k^T s_k / \|y_k\|^2$ , care este pozitiv dacă se utilizează condițiile Wolfe de căutare liniară.

În MINFI, componentele cu semn schimbat ale gradientului sunt scalate ca în (15) cu diferiți factori pozitivi. Aceasta este mai avantajoasă în ceea ce privește accelerarea componentelor gradientului la zero, adică accelerarea convergenței.

În al patrulea set de experimente numerice algoritmul MINFI este comparat cu algoritmul BFGS clasic, în care inversa Hessianului este calculată ca în (5) și lungimea pasului de deplasare este determinată de condițiile Wolfe inexacte (6) și (7).

Figura 3 prezintă profilele de performanță ale acestor algoritmi referitor la timpul de calcul, unde  $n = 100$  și  $\varepsilon_g = 10^{-4}$ . (În aceste experimente am considerat probleme cu un număr mic de variabile pentru a avea un timp de calcul rezonabil pentru algoritmul BFGS.)

Observăm că MINFI este mult mai eficient și mult mai robust decât BFGS.

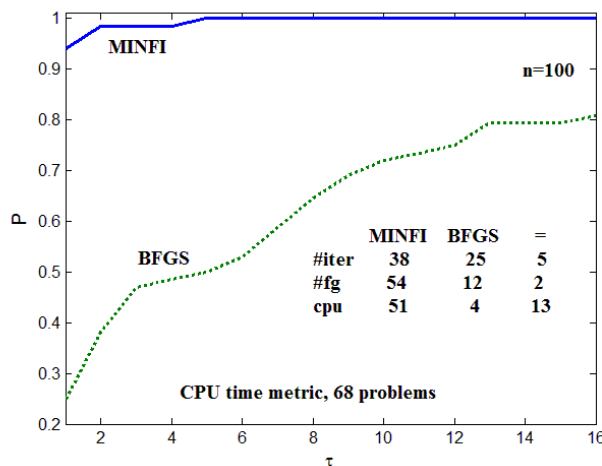


Figura 3. MINFI versus BFGS.  $n = 100$ .

## 6. Aplicații

În această secțiune vom utiliza câteva aplicații ingineresti din colecția MINPACK-2, care conține aplicații din diverse domenii de activitate (electricitate, dinamica fluidelor, elasticitate, combustie, lubrificație, teste nedistructive, cinetică chimică, etc.) (vezi Averick Carter, Moré, Xue [8]). Vom compara algoritmul MINFI versus algoritmul Cauchy cu scalarea Oren și Luenberger – COL. În experimentele noastre am considerat cinci aplicații din MINPACK-2, care sunt sumarizate în Tabelul 4. O descriere completă a acestor aplicații este dată în [8].

Tabelul 4. Aplicațiile de optimizare din MINPACK-2 considerate în acest studiu numeric.

Nr.	Numele aplicației	Parametrii
1	Torsiunea elasto-plastică a unei bare	$c=5$
2	Distribuția presiunii într-un lagăr de alunecare (cuzinet)	$ecc=0.1, b=10$
3	Proiectarea optimă a unei bare cu rigiditate torsională maximă	$\lambda = 0.008$
4	Combustia staționară a unui material solid	$\lambda = 5$
5	Suprafețe minimale cu condiții Enneper pe frontieră	-

Ultima coloană a Tabelului 4 conține valorile parametrilor asociați acestor aplicații utilizați în testele numerice. În cele ce urmează vom prezenta aceste aplicații, după care vom arăta performanțele algoritmului MINFI în comparație cu algoritmul COL în ceea ce privește rezolvarea lor.

### Aplicația A1. Torsiunea elasto-plastică a unei bare

Problema constă în determinarea câmpului de eforturi într-o bară cilindrică infinit lungă. Versiunea infinit dimensională a acestei probleme este următoarea.

$$\min \{q(v) : v \in K\},$$

unde  $q : K \rightarrow R$  este funcția pătratică:

$$q(v) = \frac{1}{2} \int_D \|\nabla v(x)\|^2 dx - c \int_D v(x) dx$$

pentru o constantă oarecare  $c$  și  $D$  este un domeniu mărginit cu frontieră netedă. Mulțimea convexă  $K$  este definită ca:

$$K = \left\{ v \in H_0^1(D) : |v(x)| \leq \text{dist}(x, \partial D), x \in D \right\},$$

unde  $\text{dist}(\cdot, \partial D)$  este distanța la frontiera lui  $D$ , și  $H_0^1(D)$  este spațiul Hilbert al tuturor funcțiilor cu suport compact în  $D$  astfel încât  $v$  și  $\|\nabla v\|^2$  aparțin lui  $L^2(D)$ . Această formulare, precum și interpretarea fizică a acestei probleme este prezentată de Glowinski [28, pp.41-55]. Aproximarea prin elemente finite a problemei se obține prin discretizarea lui  $D$  și înlocuirea problemei de minimizare a lui  $q$  pe  $H_0^1(D)$  prin minimizarea lui  $q$  pe mulțimea funcțiilor liniare pe porțiuni care satisfac restricțiile specificate de  $K$ , așa cum este descris de Averick, Carter și Moré [9, pp.21-23]. Mai exact aproximarea prin elemente finite este definită de funcția pătratică  $q$  în forma generală:

$$q(v) = \frac{1}{2} \int_D w_q(x) \|\nabla v(x)\|^2 dx - \int_D w_l(x) v(x) dx,$$

unde  $w_q : D \rightarrow R$  și  $w_l : D \rightarrow R$  sunt funcții definite pe dreptunghiul  $D$ . În problema torsiunii  $w_q = 1$  și  $w_l = c$ .

Această problemă se rezolvă prin discretizarea lui  $D$  prin alegerea unei latice de  $n_x \times n_y$  puncte din interiorul lui  $D$ . Fie  $D = (\xi_{1,l}, \xi_{1,u}) \times (\xi_{2,l}, \xi_{2,u})$  din  $R^2$ . Nodurile  $z_{i,j} \in R^2$  pentru discretizarea dreptunghiului se obțin prin alegerea pașilor de discretizare  $h_x$  și  $h_y$  și definirea punctelor rețelei ca:

$$z_{i,j} = (\xi_{1,l} + ih_x, \xi_{2,l} + jh_y), \quad 0 \leq i \leq n_x + 1, \quad 0 \leq j \leq n_y + 1$$

astfel încât  $z_{n_x+1, n_y+1} = (\xi_{1,u}, \xi_{2,u})$ . Discretizarea constă din triunghiurile elementare  $T_L$  cu vârfurile în nodurile  $z_{i,j}$ ,  $z_{i+1,j}$  și  $z_{i,j+1}$ , precum și din triunghiurile elementare  $T_U$  cu vârfurile în nodurile  $z_{i,j}$ ,  $z_{i-1,j}$  și  $z_{i,j-1}$ . Cu acestea, o aproximare a problemei torsiunii se obține prin minimizarea lui  $q$  peste spațiul funcțiilor liniare pe porțiuni  $v$  care iau valorile  $v_{i,j}$  în punctele  $z_{i,j}$ . Aproximarea integralei

$$\int_D w_q(x) \|\nabla v(x)\|^2 dx$$

peste elementul  $T_L$  este funcția pătratică  $q_{i,j}^L(v)$ , unde

$$q_{i,j}^L(v) = \mu_{i,j} \left\{ \left( \frac{v_{i+1,j} - v_{i,j}}{h_x} \right)^2 + \left( \frac{v_{i,j+1} - v_{i,j}}{h_y} \right)^2 \right\},$$

$$\mu_{i,j} = \frac{h_x h_y}{6} \left\{ w_q(z_{i,j}) + w_q(z_{i+1,j}) + w_q(z_{i,j+1}) \right\}.$$

În mod similar, aproximarea peste elementul  $T_U$  este funcția pătratică  $q_{i,j}^U(v)$ , unde

$$q_{i,j}^U(v) = \lambda_{i,j} \left\{ \left( \frac{v_{i-1,j} - v_{i,j}}{h_x} \right)^2 + \left( \frac{v_{i,j-1} - v_{i,j}}{h_y} \right)^2 \right\},$$

$$\lambda_{i,j} = \frac{h_x h_y}{6} \left\{ w_q(z_{i,j}) + w_q(z_{i-1,j}) + w_q(z_{i,j-1}) \right\}.$$

Deci, aproximarea prin elemente finite a problemei conduce la următoarea problemă de programare pătratică:

$$\min \{q(v) : v \in \Omega\},$$

unde  $q$  este funcția pătratică

$$q(v) = \frac{1}{2} \sum (q_{i,j}^L(v) + q_{i,j}^U(v)) - h_x h_y \sum w_l(z_{i,j}) v_{i,j}.$$

În această formulare  $q_{i,j}^L$  este definită numai pentru  $0 \leq i \leq n_x$  și  $0 \leq j \leq n_y$ , în timp ce  $q_{i,j}^U$  este definită pentru  $1 \leq i \leq n_x + 1$  și  $1 \leq j \leq n_y + 1$ . De asemenea pentru această problemă  $w_q = 1$  și  $w_l = c$ , iar domeniul de admisibilitate este mulțimea  $\Omega$ , unde  $\Omega = \{v \in R^{n_x n_y} : |v_{i,j}| \leq d_{i,j}\}$ , unde  $d_{i,j}$  este valoarea lui  $\text{dist}(\cdot, \partial D)$  în nodul  $z_{i,j}$ .

În experimentele noastre am considerat  $D = (0,1) \times (0,1)$  și  $c = 5$ . Rezultate numerice privind această problemă sunt prezentate de exemplu în Elliot și Ockendon [23], O'Leary și Yang [41] și Moré și Toraldo [38]. Pentru  $n_x = 200, n_y = 200$  soluția problemei este ilustrată în Figura 4.

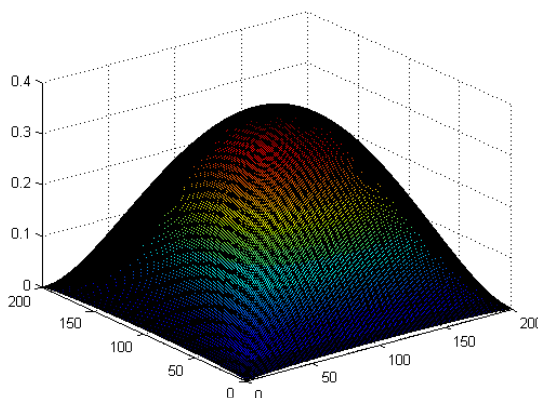


Figura 4. Soluția aplicației A1.  $n_x = 200, n_y = 200$

#### Aplicația A2. Distribuția presiunii într-un lagăr de alunecare (cuzinet)

Problema constă în determinarea distribuției presiunii într-un film subțire de lubrifiant între doi cilindri circulari. Versiunea infinit-dimensională a problemei este:

$$\min \{q(v) : v \in K\},$$

$$q(v) = \frac{1}{2} \int_D w_q(x) \|\nabla v(x)\|^2 dx - \int_D w_l(x) v(x) dx$$

Cu

$$w_q(z_1, z_2) = (1 + \varepsilon \cos z_1)^3, \quad w_l(z_1, z_2) = \varepsilon \sin z_1,$$

pentru o anumită constantă  $\varepsilon \in (0,1)$  și  $D = (0, 2\pi) \times (0, 2b)$  unde  $b > 0$  este o constantă. Mulțimea convexă  $K$  este  $K = \{v \in H_0^1(D) : v \in D, v \geq 0\}$ . Aproximarea prin elemente finite a acestei probleme se obține exact ca în problema de mai sus, unde de data aceasta  $w_q(\xi_1, \xi_2) = (1 + \varepsilon \cos \xi_1)^3$  și  $w_l(\xi_1, \xi_2) = \varepsilon \sin \xi_1$ . Domeniul de admisibilitate este mulțimea

$$\Omega = \{v \in R^{n_x n_y} : v_{i,j} \geq 0\}.$$

În experimentele noastre am considerat  $D = (0, 2\pi) \times (0, 2b)$ ,  $b = 10$  și  $\varepsilon = 0.1$ . Rezultate numerice privind această problemă se găsesc în Lin și Cryer [36], Cimatti și Menchi [18] și Moré și Toraldo [38]. Figura 5 ilustrează soluția problemei pentru  $nx = 200$ ,  $ny = 200$ .

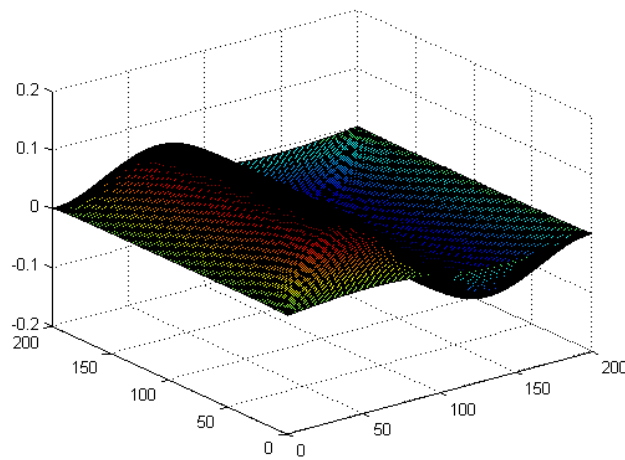


Figura 5. Soluția aplicației A2.  $nx = 200$ ,  $ny = 200$

### Aplicația A3. Proiectarea optimă a unei bare cu rigiditate torsională maximă

Problema constă în a determina în mod optim plasarea a două materiale elastice în secțiunea transversală a unei bare cu rigiditate torsională maximă. Formularea problemei se găsește în Goodman, Kohn și Reyna [30] și Averick, Carter și Moré [9].

Fie  $D \subset \mathbb{R}^2$  un domeniu mărginit și  $w < |D|$ , unde  $|D|$  reprezintă aria lui  $D$ . Problema se formulează sub forma:

$$\min \{ F(v, \Omega) : v \in H_0^1(D), |\Omega| = w \},$$

unde

$$F(v, \Omega) = \int_D \left\{ \frac{1}{2} \mu(x) \|\nabla v(x)\|^2 + v(x) \right\} dx,$$

și

$$\mu(x) = \mu_1 \text{ pentru } x \in \Omega, \text{ și } \mu(x) = \mu_2 \text{ pentru } x \notin \Omega.$$

Inversele constantelor  $\mu_1$  și  $\mu_2$  sunt modulele de elasticitate în bară. Se presupune că  $\mu_1 < \mu_2$ . Goodman, Kohn și Reyna [30] dau detalii asupra acestei probleme și o formulează în termenii proiectării optime a unei familii de probleme de optimizare de forma:

$$\min \{ f_\lambda(v) : v \in H_0^1(D) \},$$

unde  $f_\lambda : H_0^1(D) \rightarrow \mathbb{R}$  este funcționala

$$f_\lambda(v) = \int_D \left\{ \psi_\lambda(\|\nabla v(x)\|) + v(x) \right\} dx$$

și  $\psi_\lambda : \mathbb{R} \rightarrow \mathbb{R}$  este o funcție pătratică pe porțiuni. În această formulare  $\lambda$  este multiplicatorul Lagrange asociat problemei, iar funcția pătratică pe porțiuni  $\psi_\lambda$  este de forma:



$$\psi_\lambda(t) = \begin{cases} \frac{1}{2}\mu_2 t^2, & 0 \leq t \leq t_1, \\ \mu_2 t_1(t - \frac{1}{2}t_1), & t_1 \leq t \leq t_2, \\ \frac{1}{2}\mu_1(t^2 - t_2^2) + \mu_2 t_1(t_2 - \frac{1}{2}t_1), & t_2 \leq t, \end{cases}$$

cu punctele de discontinuitate  $t_1$  și  $t_2$  definite de:

$$t_1 = \left(2\lambda \frac{\mu_1}{\mu_2}\right)^{1/2} \quad \text{și} \quad t_2 = \left(2\lambda \frac{\mu_2}{\mu_1}\right)^{1/2}.$$

Definiția punctelor de discontinuitate implică  $\mu_1 t_2 = \mu_2 t_1$ , care asigură diferențiabilitatea continuă a lui  $\psi$ . Problema considerată în Averick, Carter și Moré [9] este aceea de minimizare a lui  $f_\lambda$  pentru o valoare fixată a lui  $\lambda$ . În aceste experimente numerice se consideră  $\mu_1 = 1$  și  $\mu_2 = 2$ , astfel încât  $t_1^2 = \lambda$  și  $t_2^2 = 2\lambda$ . Într-o manieră canonică se poate obține o aproximare prin elemente finite în sensul minimizării lui  $f_\lambda$  peste spațiul funcțiilor liniare pe porțiuni  $v$  cu valorile  $v_{ij}$  în  $z_{ij}$ , unde  $z_{ij} \in \mathbb{R}^2$  sunt nodurile unei discretizări ale lui  $D$  cu pașii de discretizare  $h_x$  și  $h_y$ . Valorile  $v_{i,j}$  sunt obținute ca soluție a următoarei probleme de minimizare:

$$\min \left\{ \sum \left( f_{i,j}^L(v) + f_{i,j}^U(v) + v_{i,j} \right) : v \in \mathbb{R}^n \right\},$$

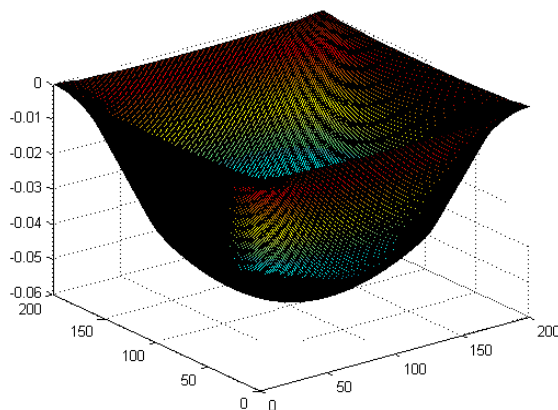
unde funcțiile  $f_{i,j}^L$  și  $f_{i,j}^U$  sunt definite ca:

$$f_{i,j}^L(v) = \frac{h_x h_y}{2} \psi_\lambda \left( d_{i,j}^+(v) \right), \quad f_{i,j}^U(v) = \frac{h_x h_y}{2} \psi_\lambda \left( d_{i,j}^-(v) \right),$$

în care

$$d_{i,j}^\pm(v) = \left\{ \left( \frac{v_{i\pm 1,j} - v_{i,j}}{h_x} \right)^2 + \left( \frac{v_{i,j\pm 1} - v_{i,j}}{h_y} \right)^2 \right\}^{1/2}.$$

În această formulare  $f_{i,j}^L$  este definită numai pentru  $0 \leq i \leq n_x$  și  $0 \leq j \leq n_y$ , în timp ce  $f_{i,j}^U$  este definită pentru  $1 \leq i \leq n_x + 1$  și  $1 \leq j \leq n_y + 1$ . Figura 6 prezintă soluția aplicației pentru discretizarea  $n_x = 200$ ,  $n_y = 200$ , cu  $\mu_1 = 1$  și  $\mu_2 = 2$



**Figura 6.** Soluția aplicației A3.  $n_x = 200$ ,  $n_y = 200$

**Aplicația A4. Combustia staționară a unui material solid**

Studiul regimului staționar al combustiei solidelor se poate exprima ca următoarea problemă de optimizare infinit dimensională.

$$\min \{ f_\lambda(v) : v \in H_0^1(D) \},$$

unde  $f_\lambda : H_0^1(D) \rightarrow R$  este funcționala

$$f_\lambda(v) = \int_D \left\{ \frac{1}{2} \|\nabla v(x)\|^2 - \lambda \exp[v(x)] \right\} dx,$$

și  $\lambda \geq 0$  un parametru cunoscut. Această problemă este formularea variațională a următoarei probleme cu valori la limită (problema variațională Bratu):

$$-\Delta v(x) = \lambda \exp[v(x)], \quad x \in D, \quad v(x) = 0 \text{ pentru } x \in \partial D$$

unde  $\Delta$  este operatorul Laplacian. Aris [7], și Bebernes și Eberly [13] discută această problemă în contextul combustiei solidelor. O proprietate interesantă a acestei probleme este că  $f_\lambda$  este nemărginită inferior pentru orice  $\lambda > 0$ . Există totuși o valoare  $\lambda_{FK} > 0$ , astfel încât pentru  $\lambda \in [0, \lambda_{FK}]$  funcția  $f_\lambda$  are un minim unic, dar nu are nici un minim pentru  $\lambda > \lambda_{FK}$ . Dacă  $D$  este pătratul unitar, atunci  $\lambda_{FK} = 6.81$ , cunoscut ca parametrul Frank-Kamenetskii.

Problema este rezolvată utilizând aproximarea prin elemente finite, prin minimizarea lui  $f$  peste spațiul funcțiilor liniare pe porțiuni  $v$  cu valorile  $v_{ij}$  în  $z_{ij}$ , unde  $z_{ij} \in R^2$  sunt nodurile unei discretizări ale lui  $D$  cu pașii  $h_x$  și respectiv  $h_y$ . Valorile  $v_{ij}$  sunt obținute ca soluții ale următoare probleme de minimizare:

$$\min \left\{ \sum (f_{ij}^L(v) + f_{ij}^U(v)) : v \in R^n \right\},$$

unde

$$f_{ij}^L(v) = \frac{h_x h_y}{4} \left\{ \left( \frac{v_{i+1,j} - v_{ij}}{h_x} \right)^2 + \left( \frac{v_{i,j+1} - v_{ij}}{h_y} \right)^2 - \lambda \mu_{ij}^L \right\},$$

$$\mu_{ij}^L = \frac{2}{3} \left\{ \exp(v_{ij}) + \exp(v_{i+1,j}) + \exp(v_{i,j+1}) \right\},$$

$$f_{ij}^U(v) = \frac{h_x h_y}{4} \left\{ \left( \frac{v_{i-1,j} - v_{ij}}{h_x} \right)^2 + \left( \frac{v_{i,j-1} - v_{ij}}{h_y} \right)^2 - \lambda \mu_{ij}^U \right\},$$

$$\mu_{ij}^U = \frac{2}{3} \left\{ \exp(v_{ij}) + \exp(v_{i-1,j}) + \exp(v_{i,j-1}) \right\}.$$

În această formulare  $f_{ij}^L$  este definit numai când  $0 \leq i \leq n_x$  și  $0 \leq j \leq n_y$ , în timp ce  $f_{ij}^U$  este definit când  $1 \leq i \leq n_x + 1$  și  $1 \leq j \leq n_y + 1$ .

Figura 7 arată soluția problemei combustiei solidelor (problema variațională Bratu) pentru  $\lambda = 5$ ,  $n_x = 200$  și  $n_y = 200$ .

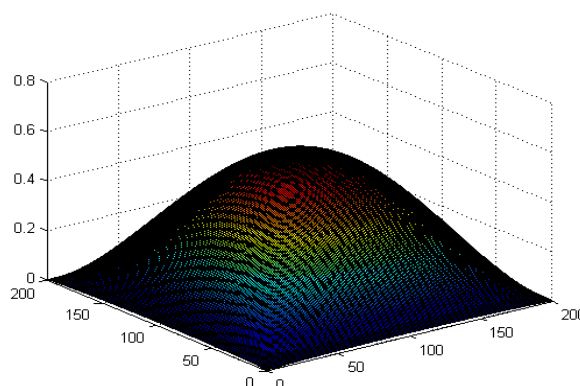


Figura 7. Soluția aplicației A4.  $n_x = 200$ ,  $n_y = 200$

### Aplicația A5. Suprafețe minimale

Determinarea suprafeței cu arie minimă când condițiile pe frontiera unui domeniu convex  $D$  sunt cunoscute este o problemă infinit dimensională de forma:

$$\min \{ f(v) : v \in K \},$$

unde  $f : K \rightarrow R$  este funcționala

$$f(v) = \int_D \left( 1 + \|\nabla v(x)\|^2 \right)^{1/2} dx$$

și mulțimea  $K$  este definită prin:

$$K = \{ v \in H^1(D) : v(x) = v_D(x), x \in \partial D \}$$

pentru o anumită funcție  $v_D : \partial D \rightarrow R$ . Funcția  $v_D$  definește în mod unic soluția problemei suprafeței minimale [40]. O descriere a acestei probleme se găsește de asemenea în [15, pagina 128].

O suprafață minimă interesantă a fost descoperită de Enneper prin definirea lui  $v_D$  pe domeniul  $D = (-1/2, 1/2) \times (-1/2, 1/2)$  cu  $v_D(\xi_1, \xi_2) = u^2 - v^2$ , unde  $u$  și  $v$  sunt soluțiile unice ale ecuațiilor

$$\xi_1 = u + uv^2 - u^3/3, \quad \xi_2 = -v - u^2v + v^3/3.$$

O aproximație prin elemente finite a acestei probleme se obține prin minimizarea lui  $f$  pe spațiul funcțiilor liniare pe porțiuni  $v$  cu valorile  $v_{i,j}$  în  $z_{i,j}$ , unde  $z_{i,j} \in R^2$  sunt nodurile unei triangulații a lui  $D$  cu pașii de discretizare  $h_x$  și  $h_y$ . Valorile  $v_{i,j}$  sunt obținute prin rezolvarea următoarei probleme de minimizare:

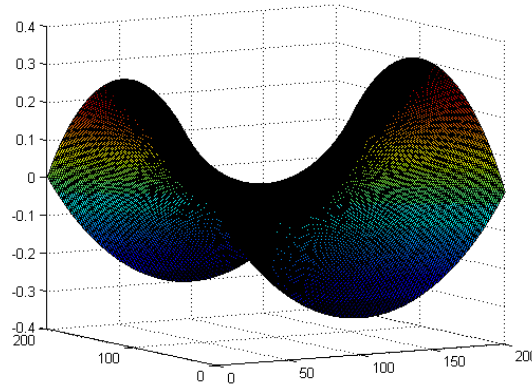
$$\min \left\{ \sum (f_{i,j}^L(v) + f_{i,j}^U(v)) : v \in R^n \right\}$$

unde funcțiile  $f_{i,j}^L$  și  $f_{i,j}^U$  sunt definite ca:

$$f_{i,j}^L(v) = \frac{h_x h_y}{2} \left\{ 1 + \left( \frac{v_{i+1,j} - v_{i,j}}{h_x} \right)^2 + \left( \frac{v_{i,j+1} - v_{i,j}}{h_y} \right)^2 \right\}^{1/2},$$

$$f_{i,j}^U(v) = \frac{h_x h_y}{2} \left\{ 1 + \left( \frac{v_{i-1,j} - v_{i,j}}{h_x} \right)^2 + \left( \frac{v_{i,j-1} - v_{i,j}}{h_y} \right)^2 \right\}^{1/2}.$$

În această formulare,  $f_{i,j}^L$  este definită numai pentru  $0 \leq i \leq n_x$  și  $0 \leq j \leq n_y$ , în timp ce  $f_{i,j}^U$  este definită pentru  $1 \leq i \leq n_x + 1$  și  $1 \leq j \leq n_y + 1$ . O reprezentare a suprafeței cu arie minimă în aceste condiții pe frontieră este ilustrată în Figura 8.



**Figura 8.** Soluția aplicației A5.  $n_x = 200$ ,  $n_y = 200$

Tabelele 5 și 6 conțin performanțele algoritmilor MINFI și COL pentru rezolvarea acestor aplicații, unde numărul de variabile este  $n = 10000$  și respectiv  $n = 40000$ , iar  $\varepsilon_g = 10^{-5}$  în criteriul de oprire a iterațiilor. În ambii algoritmi, lungimea pasului a fost calculată utilizând condițiile de căutare liniară Wolfe date de (6) și (7). În aceste tabele #iter este numărul de iterații, iar cpu este timpul de calcul este exprimat în secunde.

**Tabelul 5.** Performanțele lui MINFI și COL pentru rezolvarea a 5 aplicații MINPACK-2.  $n = 10000$

Nr.	MINFI		COL	
	#iter	cpu	#iter	cpu
1	1587	3.27	1922	3.55
2	11309	26.32	14685	30.69
3	9927	33.81	16130	51.47
4	7342	55.50	7390	54.23
5	3044	8.18	5096	12.66
Total	33209	127.08	45223	152.60

**Tabelul 6.** Performanțele lui MINFI și COL pentru rezolvarea a 5 aplicații MINPACK-2.  $n = 40000$

Nr.	MINFI		COL	
	#iter	cpu	#iter	cpu
1	4925	40.32	4420	32.45
2	33950	307.63	48091	393.67
3	19731	266.48	28878	366.20
4	11882	357.76	14463	423.56
5	9193	97.93	16495	162.30
Total	79681	1070.12	112347	1378.18

Referitor la numărul de iterații cât și la timpul de calcul, din aceste tabele vedem că MINFI este cel mai bun față de COL.

**Discuție.** Toate rezultatele numerice obținute până acum utilizează o valoare a multiplicatorului Lagrange  $\lambda$  din intervalul  $(r, +\infty)$  determinată prin procedura adaptivă (22), luând în considerare condiția de conjugare (20). Chiar dacă această abordare bazată pe condiția de conjugare este cumva stranie, totuși experimentele numerice arată că aceasta este benefică în cadrul algoritmului MINFI. Totuși, valoarea multiplicatorului Lagrange nu este sub nicio formă o soluție a ecuației  $F(\lambda) = 0$ , unde  $F(\lambda)$  este dat de (16), așa cum ar trebui să fie conform teoriei optimizării cu restricții [4]. Procedura adaptivă (22) furnizează numai o valoare a lui  $\lambda$  din intervalul  $(r, +\infty)$ .

Pentru a vedea comportarea algoritmului MINFI, în cele ce urmează vom considera o procedură bazată pe căutarea unei soluții a ecuației  $F(\lambda) = 0$  în intervalul  $(r, +\infty)$ , utilizând de exemplu metoda Newton-Raphson. În acest caz metoda Newton-Raphson calculează aproximații ale multiplicatorului Lagrange prin formula iterativă:

$$\lambda_{j+1} = \lambda_j - \frac{F(\lambda_j)}{F'(\lambda_j)}, \quad j = 0, 1, \dots, \quad (40)$$

cu  $\lambda_0 = r + \theta$  până când  $|F(\lambda_{j+1})| \leq 10^{-2}$ . Tabelul 7 prezintă performanțele algoritmului MINFI cu metoda Newton-Raphson pentru rezolvarea aplicațiilor din colecția MINPACK-2 considerată în studiul numeric de mai sus cu  $n = 10000$  și respectiv  $n = 40000$ .

**Tabelul 7.** Performanțele lui MINFI cu Newton Raphson pentru rezolvarea a 5 aplicații MINPACK-2 .

	MINFI		MINFI	
	$n = 10000$		$n = 40000$	
Nr.	#iter	cpu	#iter	cpu
1	1739	6.20	4977	70.80
2	11611	34.61	34418	399.93
3	8940	36.92	19207	312.46
4	6608	53.60	14705	474.00
5	3105	10.34	8698	115.17
Total	32003	141.67	82005	1372.36

Din Tabelele 5, 6 și 7 vedem că utilizarea algoritmului Newton-Raphson inițializat în  $\lambda_0 = r + \theta$  ( $\theta = 1$ ) nu reprezintă o îmbunătățire a lui MINFI. Referitor la timpul de calcul vedem că MINFI cu Newton-Raphson pentru calculul multiplicatorului Lagrange necesită 141.67 secunde pentru aplicațiile din MINPACK-2 cu 10000 variabile, și 1372.36 secunde pentru aplicațiile cu 40000 variabile. Pe de altă parte, MINFI cu procedura adaptivă (22) bazată pe condiția de conjugare (20) necesită numai 127.08 secunde pentru aplicațiile cu 10000 variabile și respective 1070.12 secunde pentru cele cu 40000 variabile. Ca o concluzie, procedura adaptivă (22) bazată pe condiția de conjugare (20) reprezintă o foarte bună procedură de estimare a multiplicatorului Lagrange din algoritmul de minimizare bazat pe aproximarea diagonală a Hessianului cu minimizarea funcției măsură a lui Byrd și Nocedal.

## 5. Concluzii

Noutatea metodei prezentată în acest articol constă în faptul că elementele diagonale ale matricei diagonale care înmulțește gradientul cu semn schimbat sunt determinate prin minimizarea funcției măsură a lui Byrd și Nocedal referitor la ecuația secantei slabă a lui Dennis și Wolkowicz. Algoritmul este simplu. Totuși, complicațiile apar când trebuie determinată o valoare a multiplicatorului Lagrange corespunzător problemei de minimizare a funcției măsură referitor la ecuația secantei slabă. Pentru a evita aceste complicații (date de complexitatea funcției  $F(\lambda)$ ) am introdus și utilizat condiția de conjugare proprie algoritmilor de gradient conjugat, precum și o schemă ite-

rativă bazată pe polul maxim al funcției  $F(\lambda)$ . Convergența algoritmului este bazată pe operatorii trace și determinant a matricei de iterație, care este numai liniară. Experimentele computaționale arată că acest mod de abordare care implică funcția măsură Byrd-Nocedal este superior algoritmului pasului descendent, algoritmului Cauchy cu scalarea Oren și Luenberger, precum și algoritmului BFGS pentru clase mari de probleme de optimizare fără restricții cu diverse structuri și complexități, incluzând aplicații complexe din colecția MINPACK-2. Performanțele algoritmului sunt comparabile cu cele ale algoritmului lui Barzilai și Borwein. Ceea ce relevă această abordare, bazată pe aproximarea diagonală a matricei Hessian a funcției de minimizat, în care elementele diagonale se determină prin minimizarea funcției măsură a lui Byrd și Nocedal referitor la ecuația secantei slabă Dennis-Wolkowicz, este faptul că algoritmi quasi-Newton nu sunt foarte tare dependenți de aproximarea Hessianului funcției de minimizat. Observăm că o matrice diagonală (cea mai simplă matrice) poate înlocui cu succes aproximarea  $n \times n$  – dimensională a Hessianului așa cum este de exemplu utilizată în metoda BFGS. Evident, prețul pe care trebuie să-l plătim este convergența liniară. Ca o observație finală, menționăm că funcția măsură Byrd-Nocedal nu este numai un ingredient teoretic foarte important în analiza metodelor quasi-Newton, dar, după cum am arătat, aceasta poate conduce la proiectarea de algoritmi eficienți și robuști de optimizare fără restricții (vezi [6]).

## BIBLIOGRAFIE

1. Andrei N. An unconstrained optimization test functions collection, *Advanced Modeling and Optimization – An Electronic International Journal* 2008;10:147-161;
2. Andrei N. An adaptive conjugate gradient algorithm for large-scale unconstrained optimization. *Journal of Computational and Applied Mathematics* 2016;292:83-91;
3. Andrei N. Eigenvalues versus singular values study in conjugate gradient algorithms for large-scale unconstrained optimization, *Optimization Methods and Software* 2017;32(3): 534-551;
4. Andrei N. *Continuous Nonlinear Optimization for Engineering Applications in GAMS Technology*. Springer Optimization and Its Applications 121. Springer, Berlin, 2017;
5. Andrei N. A new three-term conjugate gradient algorithm for unconstrained optimization. *Numerical Algorithms* 2015;68:305-321;
6. Andrei, N., A diagonal quasi-Newton updating method based on minimizing the measure function of Byrd and Nocedal for unconstrained optimization. *Optimization*, DOI: 10.1080/02331934.2018.1482298;
7. Aris, R., *The mathematical theory of diffusion and reaction in permeable catalysis*. Oxford, 1975;
8. Averick BM, Carter RG, Moré JJ, Xue G. The MINPACK-2 test problem collection. Preprint MCS-P153-0692, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1992;
9. Averick, B.M., Carter, R.G., Moré, J.J., The MINPACK-2 test problem collection (Preliminary version), Mathematics and Computer Science Division, Argonne National Laboratory, Technical Memorandum No. 150, May 1991;
10. Babaie-Kafaki S. On optimality of the parameters of self-scaling memoryless quasi-Newton updating formulae. *Journal of Optimization Theory and Applications* 2015;167(1):91-101;
11. Babaie-Kafaki S. A modified scaling parameter for the memoryless BFGS updating formula. *Numerical Algorithms* 2016;72(2):425-433;
12. Barzilai, J., Borwein, J.M., Two-point step size gradient methods. *IMA Journal of Numerical Analysis* 1988;8:141-148;
13. Bebernes, J., Eberly, D., *Mathematical problems from combustion theory*. Applied Mathematical Sciences 83, Springer-Verlag, 1989;
14. Bertsekas D.P. *Nonlinear Programming*. Athena Scientific, Belmont, Massachusetts, 1995;
15. Boyd, S., Vandenberghe, L., *Convex Optimization*. Cambridge University Press, Cambridge, 2004;

16. Broyden CG. The convergence of a class of double-rank minimization algorithms. I. General considerations, *J. Inst. Math. Appl.* 1970;6:76-90;
17. Byrd R, Nocedal J. A tool for the analysis of quasi-Newton methods with application to unconstrained minimization. *SIAM Journal on Numerical Analysis* 1989;26:727-739;
18. Cimatti, G., Menchi, O., On the numerical solution of a variational inequality connected with the hydrodynamic lubrication of a complete journal bearing. *Calcolo*, 15, (1978), pp.249-258;
19. Davidon WC. Variable metric method for minimization. Technical Report ANL-5990 (revised), Argonne National Laboratory, Argonne, IL, 1959;
20. Dennis JE, Schnabel RB, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice Hall Series in Computational Mathematics, Prentice Hall, Englewood Cliffs, NJ, 1983;
21. Dennis JE, Wolkowicz H. Sizing and least-change secant methods, *SIAM J. Numerical Analysis*, 1993;30(5):1291-1314;
22. Dolan ED, Moré JJ. Benchmarking optimization software with performance profiles, *Mathematical Programming* 2002;91:201-213;
23. Elliot, C.M., Ockendon, J.R., *Weak and variational methods for moving boundary problems*. Research Notes in Mathematics, vol.59, Pittman, 1982;
24. Farid M, Leong WJ, Zheng L. A new diagonal gradient-type method for large scale unconstrained optimization. *U.P.B. Sci. Bull., Series A*, 2013;75(1):57-64;
25. Fletcher R, Powell MJD. A rapidly convergent descent method for minimization. *Computer Journal*, 1963;6:163-168;
26. Fletcher R. A new approach to variable metric algorithms, *The Computer Journal* 1970;13: 317-322;
27. Fletcher R. A new variational result for quasi-Newton formulae. *SIAM Journal on Optimization* 1991;1:18-21;
28. Glowinski, R., *Numerical Methods for Nonlinear Variational Problems*. Springer-Verlag, Berlin, 1984;
29. Goldfarb D. A family of variable metric methods derived by variation mean, *Mathematics of Computation* 1970;23:23-26;
30. Goodman, J., Kohn, R., Reyna, L., Numerical study of a relaxed variational problem from optimal design. *Comput. Methods Appl. Mech. Engrg.*, 57, 1986, pp.107-127;
31. Hassan MA, Leong WJ, Farid M. A new gradient method via quasi-Cauchy relation which guarantees descent. *J. Comput. Appl. Math.* 2009;230(1):300-305;
32. Huang, S., Wan, Z., Zhang, J. An extended nonmonotone line search technique for large-scale unconstrained optimization. *Journal of Computational and Applied Mathematics* 2018;330:586-604;
33. Leong WJ, Hassan MA, Farid M. A monotone gradient method via weak secant equation for unconstrained optimization. *Taiwanese J. Math.* 2010;14(2):413-423;
34. Leong WJ, Farid M, Hassan MA. Improved Hessian approximation with modified quasi-Cauchy relation for a gradient-type method. *AMO—Advanced Modeling and Optimization*, 2010;12(1):37-44;
35. Leong WJ, Farid M, Hassan MA. Scaling on diagonal quasi-Newton update for large-scale unconstrained optimization. *Bull. Malays. Math. Sci. Soc.* 2012;35(2):247-256;
36. Lin, Y., Cryer, C.W., An alternating direction implicit algorithm for the solution of linear complementarity problems arising from free boundary problems. *Appl. Math. Optim.*, 13, 1985, pp.1-17;
37. Luenberger, DG, Ye, Y. *Linear and Nonlinear Programming*. Fourth Edition. Springer, International Series in Operations Research & Management Science. New York, 2016;

38. Moré, J.J., Toraldo, G., On the solution of large quadratic programming problems with bound constraints. *SIAM J. Optimization*, 1, 1991, pp.93-113;
39. Nazareth J.L., If quasi-Newton then why not quasi-Cauchy? *SIAG/Opt Views-and-news* 1995; 6:11-14;
40. Nitsche, J.C.C., *Lectures on minimal surfaces. Vol.1*, Cambridge University Press, 1989;
41. O'Leary, D.P., Yang, W.H., Elastoplastic torsion by quadratic programming, *Comput. Methods Appl. Mech. Engrg.*, 16, 1978, pp.361-368;
42. Oren SS, Luenberger DG. Self-scaling variable metric (SSVM) algorithms. *Management Science*. 1974;20(5):845-862;
43. Shanno DF. Conditioning of quasi-Newton methods for function minimization, *Mathematics of Computation* 1970;24:647-656;
44. Wan, Z., Teo, KL., Shen, XL. New BFGS method for unconstrained optimization problem based on modified Armijo line search. *Optimization: A Journal of Mathematical Programming and Operations Research* 2014;63(2):285-304;
45. Wan, Z., Chen, Y., Huang, S., DongFeng, D. A modified nonmonotone BFGS algorithm for solving smooth nonlinear equations. *Optimization Letters* 2014;8:1845-1860;
46. Wolfe P. Convergence conditions for ascent methods. *SIAM Review*, 1969;11:226-235;
47. Wolfe P. Convergence conditions for ascent methods. II: Some corrections. *SIAM Review* 1971;13:185-188;
48. Zhu M, Nazareth JL, Wolkowicz H. The quasi-Cauchy relation and diagonal updating. *SIAM Journal on Optimization* 1999;9(4):1192-1204;
49. Zoutendijk G. Nonlinear programming, computational methods. In J. Abadie (Ed.) *Integer and Nonlinear Programming*, North-Holland, Amsterdam, 1970;37-86.



**Neculai ANDREI** este cercetător științific gradul I în ICI București de mai bine de 45 de ani. Autor a peste 400 de lucrări științifice, publicate în reviste cotate ISI și a peste 20 de cărți publicate în edituri recunoscute, câteva dintre acestea în Editura Springer, principalele realizări științifice sunt în domeniile: calcul numeric de înaltă performanță, tehnologia matricelor rare, sisteme liniar-dinamice de mari dimensiuni, modelare matematică limbaje de modelare orientate algebric și compilatoare, optimizare liniară și neliniară de mari dimensiuni. Este membru în colegiile de redacție a trei reviste științifice cotate ISI și membru titular al AOSR.

**Neculai ANDREI** is a senior scientific researcher, working in ICI more than 45 years. Author of more than 400 scientific papers, published in highly respected ISI journals, and over 20 books, some of them published in Springer Publishing House, his main achievements are on: high performance computing, sparse matrix computation technology, large-scale linear-dynamic systems, mathematical modeling and algebraic oriented languages and compilers for mathematical programming, large-scale linear and nonlinear optimization. He is member in Editorial Boards of three ISI journals and full member of AOSR.