# COLECȚIE DE PROTOTIPURI DE MODELE DE OPTIMIZARE NELINIARĂ

Neculai Andrei

Research Institute for Informatics, Center for Advanced Modeling and Optimization, 8-10, Averescu Avenue, Bucharest 1, Romania. and

Academy of Romanian Scientists, 54, Splaiul Independentei, Bucharest 5, Romania. E-mail: nandrei@ici.ro

În această lucrare prezentăm câteva prototipuri de modele matematice de optimizare neliniară care constituie o colecție de modele de optimizare neliniară. Modelele respective sunt aplicații reale din diferite domeniile de activitate<sup>1</sup>. Contribuția noastră aici a fost în a le asambla în această colecție, a le prezenta într-un mod unitar și a le rezolva prin intermediul mai multor pachete de optimizare fără restricții.

#### 1. 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:

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

unde  $dist(.,\partial D)$  este distanța la frontiera lui D, și  $H_0^1(D)$  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 [1984, 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 and Moré [1991, pp.21-23]. Mai exact aproximarea prin elemente finite este definită de funcția pătratică q în forma generală:

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

<sup>&</sup>lt;sup>1</sup> B.M., Averick, Carter, R.G., Moré, J.J., *The MINPACK-2 test problem collection (Preliminary version)*, Mathematics and Computer Science Division, Argonne National Laboratory, Thechnical Memorandum No. 150, May 1991.

unde  $w_q: D \to R$  și  $w_l: D \to R$  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:

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

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 qpeste spațiul funcțiilor liniare pe porțiuni v care iau valorile  $v_{i,j}$  în punctele  $z_{i,j}$ . Aproximarea integralei

$$\int_{D} w_q(x) \|\nabla v(x)\|^2 \,\mathrm{d}x$$

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

$$q(v) = \frac{1}{2} \sum \left( q_{i,j}^{L}(v) + q_{i,j}^{U}(v) \right) - h_{x} h_{y} \sum w_{l}(z_{i,j}) v_{i,j}.$$

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

Problema de optimizare nu este simplă. Aceasta implică minimizarea funcției q(v) definită mai sus, pentru care trebuie să calculăm gradientul. În cele ce urmează vom prezenta rezultatele furnizate de 23 de algoritmi de gradient conjugat, de algoritmul BFGS cu memorie limitata, precum și de algoritmul Newton trunchiat (NT) [Andrei, 2007]<sup>2</sup>.

Considerând  $D = (0,1) \times (0,1)$ , c = 5 și nx = 200, ny = 200, atunci performanța algoritmilor de gradient conjugat este prezentată în tabelul A1a. Toți algoritmii utilizează același criteriu de oprire a iterațiilor  $\|\nabla f(x_k)\|_{\infty} \le 10^{-6}$ , aceeași procedură de căutare liniară dată de condițiile Wolfe și criteriul de restart Powell.

$n_{x} = 200, n_{y} = 200, \epsilon = 10$									
	#iter	#fg	сри			#iter	#fg	cpu	
HS	431	556	39.93		LS-CD	434	600	42.01	
FR	1143	1236	93.25		DL (t=1)	524	616	45.15	
PRP	500	710	49.51		DL+(t=1)	572	653	48.29	
PRP+	454	651	45.25		CONMIN	242	486	30.04	
CD	854	970	71.82		CG_DESCENT	323	647	64.31	
LS	516	712	50.04		SCG	526	652	47.27	
DY	464	488	36.79		sPRP	542	772	54.00	
TAS	515	728	50.80		sFR	1097	1202	90.65	
PRP-FR	491	704	48.93		SCALCG	414	541	33.64	
GN	438	613	42.75		ACGSD	473	505	37.98	
HS-DY	513	655	46.89		aCGSD	575	645	47.94	
hDY	451	585	41.62		PRP-SDC	474	521	38.88	

**Tabelul A1a.** Performanța algoritmilor de gradient conjugat. nr = 200, nv = 200,  $\varepsilon = 10^{-6}$ 

Algoritmul L-BFGS furnizează o soluție a căror caracteristici privind numărul de iterații (#iter), numărul de evaluări ale funcției și gradientului (#fg) și timpul de calcul (CPU) este precizat în tabelul A1b.

$n = 40000$ , $\varepsilon = 10^{-6}$ .						
#iter #fg cp						
L-BFGS (m=3)	322	338	20.90			
L-BFGS (m=5)	328	677	21.51			

340

318

L-BFGS (m=7)

L-BFGS (m=9)

Tabelul A1b. Performanța algoritmului L-BFGS.

Algoritmul TN furnizează o soluție a căror caracteristici privind numărul de iterații (#iter), numărul de evaluări ale funcției și gradientului (#fg) și timpul de calcul (CPU - secunde) este precizat în tabelul A1c.

1027

1353

22.96

Tabelul A1c. Performanța algoritmului TN.

$$\mathcal{E} = 10^{-6}$$

<sup>&</sup>lt;sup>2</sup> Neculai Andrei, *Critica Rațiunii Algoritmilor de Optimizare fără Restricții*. Manuscris predat la Editura Academiei Române, 2007, 780 pagini și CD.

	#iter	#fg	сри	
<i>n</i> = 40000	13	307	19.75	

Rezultate numerice privind această problemă sunt prezentate de exemplu în Elliot și Ockendon [1982], O'Leary și Yang [1978] și Moré și Toraldo [1991]. Pentru nx = 200, ny = 200 soluția problemei este ilustrată în figura A1a.



Fig. A1a. Soluția aplicației. nx = 200, ny = 200. 40000 variabile.

Rezultatele obținute de ACGHES sunt prezentate în Tabelul A1d și figura A1b.

Table A1d. Performanța algoritmului ACGHES.								
n	#iter	#fg	CPU (sec)	fx				

n	#iter	#fg	CPU (sec)	fx
40000	2001	2097	30.74	-0.3458351472



Fig. A1b. Soluția dată de ACGHES. nx = 200, ny = 200. c = 5. 40000 variabile.

# 2. 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  $K = \{v \in H_0^1(D) : v \in D, v \ge 0\}$ . 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

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

Considerând b = 10 și  $\varepsilon = 0.1$ , și o discretizare  $n_x \times n_y$  a domeniului  $D = (0,2\pi) \times (0,2b)$ , unde nx = 200 și ny = 200, atunci performanța algoritmilor de gradient conjugat este ilustrată în tabelul A2a.

Tabelul A2a. Performanța algoritmilor de gradient conjugat.

nx = 20	0, ny = 200.	<i>E</i> =	$=10^{-6}$ .	
				Г

	#iter	#fg	сри		#iter	#fg	сри
HS	1337	1769	129.83	LS-CD	1421	1967	141.10
FR	2001	2114	162.51	DL (t=1)	1274	1570	116.20
PRP	1242	1773	125.79	DL+(t=1)	1417	1756	130.21
PRP+	1328	1891	134.48	CONMIN	819	1657	103.86
CD	1884	2112	159.11	CG_DESCENT	794	1589	156.01
LS	938	1333	94.00	SCG	1252	1581	116.45
DY	1054	1084	83.65	sPRP	1565	2216	159.03
TAS	1280	1803	128.37	sFR	2001	2112	163.75
PRP-FR	1547	2186	156.24	SCALCG	876	1143	72.19
GN	1149	1642	116.35	ACGSD	1070	1156	88.17
HS-DY	1341	1736	126.93	aCGSD	1271	1406	108.15
hDY	1358	1748	128.04	PRP-SDC	1237	1497	111.33

Performanța algoritmului LBFGS este ilustrată în tabelul A2b.

Tabelul A2b. Performanța algoritmului L-BFGS.

$n = 40000 \cdot \varepsilon = 10^{-6} \cdot \varepsilon$						
	#iter	#fg	CPU			
L-BFGS (m=3)	898	935	58.47			
L-BFGS (m=5)	788	1750	52.15			
L-BFGS (m=7)	813	2584	55.33			
L-BFGS (m=9)	782	3386	54.94			

Algoritmul Newton trunchiat furnizează o soluție ca în tabelul A2c.

Tabelul A2c. Performanța algoritmului TN.

10-6

$\mathcal{E} = 10^{\circ}$ .							
	#iter	#fg	CPU				
n = 40000	35	777	50.25				

Rezultate numerice privind această problemă se găsesc în Lin și Cryer [1985], Cimatti și Menchi [1978] și Moré și Toraldo [1991]. Figura A2 ilustrează soluția problemei pentru nx = 200, ny = 200.

Observăm că cel mai bun algoritm este Newton trunchiat.



Fig. A2. Soluția aplicației. nx = 200, ny = 200. 40000 variabile

Rezultatele obținute cu ACGHES sunt prezentate în tabelul A2d.

Table A2d.	Rezultatele furnizate de ACGHES.
	106

$\mathcal{E} = 10^{-6}$						
 n	#iter	#fg	CPU (sec)	fx		
 40000	631	666	23.37	-0.2828924854		

Observăm că pentru această problemă algoritmul ACGHES care utilizează o aproximare prin diferențe finite a produsului Hessian / vector este superior din punctul de vedere al timpului de calcul.

#### 3. 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 [1986] și Averick, Carter și Moré [1991].

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

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

unde

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

şi

$$\mu(x) = \mu_1$$
 pentru  $x \in \Omega$ , și  $\mu(x) = \mu_2$  pentru  $x \notin \Omega$ 

Inversele constantelor  $\mu_1$  și  $\mu_2$  sunt modulele de elasticitate in bară. Se presupune că  $\mu_1 < \mu_2$ . Goodman, Kohn și Reyna [1986] dau detalii asupra acestei probleme și o formulează în termenii proiectării optime a unei familii de probleme de optimizare de forma:

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

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

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

și  $\psi_{\lambda} : R \to R$  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:

$$\psi_{\lambda}(t) = \begin{cases} \frac{1}{2}\mu_{2}t^{2}, & 0 \le t \le t_{1}, \\ \mu_{2}t_{1}(t - \frac{1}{2}t_{1}), & t_{1} \le t \le t_{2}, \\ \frac{1}{2}\mu_{1}(t^{2} - t_{2}^{2}) + \mu_{2}t_{1}(t_{2} - \frac{1}{2}t_{1}), & t_{2} \le t, \end{cases}$$

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

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

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 [1991] 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  $z_{ij} \in R^2$  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:

$$f_{i,j}^{L}(v) = \frac{h_{x}h_{y}}{2} \psi_{\lambda} \Big( d_{i,j}^{+}(v) \Big), \qquad f_{i,j}^{U}(v) = \frac{h_{x}h_{y}}{2} \psi_{\lambda} \Big( d_{i,j}^{-}(v) \Big),$$

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  $0 \le i \le n_x$  și  $0 \le j \le n_y$ , în timp ce  $f_{i,j}^{U}$  este definită pentru  $1 \le i \le n_x + 1$  și  $1 \le j \le n_y + 1$ .

Considerând  $\mu_1 = 1$  și  $\mu_2 = 2$ , în tabelul A3a se prezintă performanța algoritmilor de gradient conjugat pentru nx = 200 și ny = 200.

$nx = 200$ , $ny = 200$ , $\lambda = 0.008$ . $\varepsilon = 10^{-6}$ .									
	#iter	#fg	сри			#iter	#fg	сри	
HS	1716	2010	186.59		LS-CD	2001	2560	234.03	
FR	2001	2097	199.78		DL (t=1)	1470	1647	154.13	
PRP	2001	2681	242.00		DL+(t=1)	1648	1844	172.72	
PRP+	2001	2688	242.55		CONMIN	675	1370	114.48	
CD	1648	1794	168.36		CG_DESCENT	1036	2074	270.33	
LS	2001	2582	234.73		SCG	1756	2026	188.89	
DY	2001	2013	193.76		sPRP	2001	2678	242.36	
TAS	2001	2647	239.37		sFR	2001	2097	200.37	
PRP-FR	2001	2641	238.90		SCALCG	1413	1753	144.01	
GN	2001	2630	238.12		ACGSD	1098	1137	108.22	
HS-DY	1878	2215	205.07		aCGSD	1374	1420	135.36	
hDY	1878	2215	206.02		PRP-SDC	1088	1144	108.68	

Tabelul A3a. Performanța algoritmilor de gradient conjugat.

Algoritmul L-BFGS pentru diferire valori ale lui *m*, unde nx = 200 și ny = 200, cu parametrii  $\lambda = 0.008, \ \mu_1 = 1 \ \text{si} \ \mu_2 = 2.$ 

$n = 40000$ . $\varepsilon = 10^{-6}$ .									
	#iter #fg CPU								
L-BFGS (m=3)	1468	1482	121.10						
L-BFGS (m=5)	856	2343	71.83						
L-BFGS (m=7)	864	3210	74.30						
L-BFGS (m=9)	659	3871	57.93						

Tabelul A3b. Performanta algoritmului L-BFGS.

Algoritmul TN, pentru nx = 200 și ny = 200, cu parametrii  $\lambda = 0.008$ ,  $\mu_1 = 1$  și  $\mu_2 = 2$ furnizează o soluție descrisă în tabelul A3c.

Tabelul A3c. Performanța algoritmului TN.

$\mathcal{E} = 10^{-6}$ .								
	#iter	#fg	CPU					
<i>n</i> = 40000	56	1817	161.77					

Figura A3 prezintă soluția aplicației pentru discretizarea nx = 200, ny = 200.



Fig. A3. Soluția aplicației. nx = 200, ny = 200. 40000 variabile.

Rezultatele obținute cu ACGHES sunt prezentate în tabelul A3d.

Table A3d. Rezultatele furnizate de ACGHES.

$\mathcal{E} = 10^{-6}$ .									
n #iter #fg CPU (sec) fx									
40000	1017	1052	60.65	-0.0113812898					

Observăm că și în cazul acestei aplicații ACGHES furnizează o soluție optimă într-un timp de calcul mai bun față de LBFGS și TN. Observăm că și față de ceilalți algoritmi de gradient conjugat, ACGHES este superior. Remarcăm faptul că *nu putem preciza* dacă superioritatea algoritmului ACGHES este datorată procedurii de calcul a lui  $\beta_k$  bazată pe utilizarea produsului Hessian / vector sau a procedurii de accelerare. Cert este că accelerarea este o tehnică efectivă de îmbunătățire a comportării algoritmilor de gradient conjugat<sup>3</sup>.

# 4. Rezolvarea problemei Ginzburg-Landau

Această problemă apare în rezolvarea ecuațiilor Ginzburg-Landau pentru superconductoare neomogene formate din straturi de plumb și staniu în absența unui câmp magnetic. Formularea problemei se găsește în Garner și Benedek [1990] și Averick, Carter și Moré [1991].

Problema constă în a minimiza energia liberă Gibbs ca o funcție de temperatură. Versiunea infinit-dimensională are forma

$$\min\{f(v): v(-d) = v(d), v \in C^{1}[-d,d]\},\$$

<sup>&</sup>lt;sup>3</sup> N. Andrei, Acceleration of conjugate gradient algorithms forunconstrained optimization. ICI Technical Report, October 24, 2007.

unde 2d este lățimea materialului și f este funcția de energie Gibbs:

$$f(v) = \frac{1}{2d} \int_{-d}^{d} \left\{ \alpha(\xi) |v(\xi)|^{2} + \frac{1}{2} \beta(\xi) |v(\xi)|^{4} + \frac{\hbar^{2}}{4m} |v'(\xi)|^{2} \right\} d\xi,$$

în care funcțiile  $\alpha$  și  $\beta$  sunt constante pe porțiuni, pentru o valoare fixată a temperaturii T,  $\hbar$  este constanta lui Planck (1.05459e-27 erg-sec), și m este masa electronului (9.11e-28 grame). Funcțiile  $\alpha$  și  $\beta$  sunt constante pe anumite intervale corespunzătoare materialelor din care este făcut superconductorul (plumb și staniu). În această problemă plumbul în material corespunde intervalului  $[-d_s, d_s]$  și staniul se află în restul intervalului [-d, d]. Ca atare, funcțiile  $\alpha$  și  $\beta$  sunt definite ca:

$$\alpha(\xi) = \begin{cases} \alpha_N, & -d \leq \xi \leq -d_S, \\ \alpha_S, & -d_S < \xi \leq d_S, \\ \alpha_N, & d_S < \xi \leq d, \end{cases} \qquad \beta(\xi) = \begin{cases} \beta_N, & -d \leq \xi \leq -d_S, \\ \beta_S, & -d_S < \xi \leq d_S, \\ \beta_N, & d_S < \xi \leq d. \end{cases}$$

Constantele  $\alpha_s$  și  $\alpha_n$  sunt negative, dar  $\beta_s$  și  $\beta_n$  sunt pozitive.

Aproximarea prin elemente finite a acestei probleme se obține imediat prin minimizare lui f peste spațiul funcțiilor liniare pe porțiuni v cu valorile  $v_i$  în punctele  $t_i$ , unde

 $-d = t_1 < t_2 < \dots < t_n < t_{n+1} = d.$ 

Deci, valorile  $v_i$  sunt obținute ca soluție a următoarei probleme de minimizare:

$$min\left\{\frac{1}{2d}\sum_{i=1}^n f_i(v): v \in \mathbb{R}^n\right\},\$$

unde

$$f_{i}(v) = h_{i} \left\{ \frac{\alpha_{i}}{3} \frac{v_{i+1}^{3} - v_{i}^{3}}{v_{i+1} - v_{i}} + \frac{\beta_{i}}{10} \frac{v_{i+1}^{5} - v_{i}^{5}}{v_{i+1} - v_{i}} + \frac{\hbar^{2}}{4m} \left( \frac{v_{i+1} - v_{i}}{h_{i}} \right)^{2} \right\},$$

cu  $h_i = t_{i+1} - t_i$  lungimea subintervalului al i-lea, și constantele  $\alpha_i$  și  $\beta_i$  sunt valorile funcțiilor  $\alpha$ și  $\beta$  în subintervalul  $[t_i, t_{i+1}]$ .

Considerând d = 3.2 Å și temperatura T = 5, pentru 1000 de puncte de discretizare algoritmii de gradient conjugat furnizează rezultatele din tabelul A4a.

Tabelul A4a. Performanța algoritmilor de gradient conjugat.

n	_	1	۸ſ	)()	ç	_	1	n
n	_	1	υι	JU	C	_	1	υ

	#iter	#fg	сри	-		#iter	#fg	сри		
HS	14001	18202	86.97		LS-CD	14001	19518	91.70		
FR	14001	14996	74.08*		DL (t=1)	14001	18160	86.11		
PRP	14001	20051	94.66		DL+(t=1)	14001	18229	85.92		
PRP+	14001	20136	95.11	-	CONMIN	7419	15001	69.36		
CD	14001	15038	74.81*		CG_DESCENT	14001	22718	182.84		
LS	14001	19661	93.18		SCG	14001	17901	85.70		
DY	†				sPRP	14001	20120	91.69		
TAS	14001	19956	102.73		sFR	14001	14849	74.12*		
PRP-FR	14001	20019	93.03		SCALCG	15001	19744	91.98		
GN	14001	20019	94.64	_	ACGSD	14001	14246	71.20*		
HS-DY	14001	18252	85 77	-	aCGSD	14001	14688	74 66*		

	hDY	14001	18296	85.23	PRP-SDC	14001	18185	86.51
† DY n	u converge	(prea mult	e iterații)					

\* Algoritmii nu reușesc să reducă norma gradientului pentru a satisface criteriul de stop.

Tabelul A4b prezintă caracteristicile rezolvării cu L-BFGS, m = 3.

Tabelul A4b. Performanța algoritmului L-BFGS.

$n = 1000 \cdot \varepsilon = 10^{-6} \cdot \varepsilon$								
#iter #fg CPU								
L-BFGS (m=3)	1904	2001	9.45					

Pentru diverse valori ale lui *n* algoritmul TN converge și furnizează o soluție ca în tabelul A4c.

Tabelul A4c. Performanța algoritmului TN.									
$\mathcal{E} = 10^{-6}$ .									
#iter #fg CPU									
<i>n</i> = 1000	350	6581	33.54						
<i>n</i> = 2000	452	16730	164.89						

În figura A4 se arată soluția problemei pentru n = 1000 puncte de discretizare.



Fig. A4. Soluția aplicației. n = 1000.

Rezultatele obținute cu ACGHES sunt prezentate în tabelul A4d.

**Table A4d.** Rezultatele furnizate de ACGHES.  $\mathcal{E} = 10^{-6}$ . ♦ Colecție de prototipuri de modele matematice de optimizare ◆

n	#iter	#fg	CPU (sec)	fx
1000	14001	72948	5.95	-8456.19197525

Problema este dificilă pentru majoritatea algoritmilor de gradient conjugat.

### 5. Combustia staționară a unui material solid

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

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

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

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

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

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

unde  $\Delta$  este operatorul Laplacian. Aris [1975], și Bebernes și Eberly [1989] 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 \left(f_{ij}^{L}(v)+f_{ij}^{U}(v)\right): v \in \mathbb{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  $0 \le i \le n_x$  şi  $0 \le j \le n_y$ , în timp ce  $f_{ij}^{U}$  este definit când  $1 \le i \le n_x + 1$  şi  $1 \le j \le n_y + 1$ .

Tabelul A5a arată performanța algoritmilor de gradient conjugat pentru  $\lambda = 5$ ,  $n_x = 200$  și  $n_y = 200$ .

			1000	20	0, <i>hy</i> 200.0	10 .		
	#iter	#fg	сри			#iter	#fg	сри
HS	1219	1560	181.53		LS-CD	888	1228	140.19
FR	2001	2115	252.24		DL (t=1)	1081	1240	147.16
PRP	1136	1590	181.40		DL+(t=1)	870	997	117.50
PRP+	1126	1610	183.18		CONMIN	578	1169	122.65
CD	1355	1525	179.85		CG_DESCENT	435	871	138.29
LS	593	824	93.83		SCG	875	1088	126.55
DY	886	909	109.13		sPRP	967	1377	157.86
TAS	892	1263	143.64		sFR	2001	2097	250.97
PRP-FR	656	918	104.39		SCALCG	664	853	90.26
GN	783	1099	125.05		ACGSD	705	731	87.62
HS-DY	902	1158	133.83		aCGSD	867	926	110.56
hDY	918	1175	135.81		PRP-SDC	829	878	105.03

**Tabelul A5a.** Performanța algoritmilor de gradient conjugat. nx = 200, nv = 200.  $\varepsilon = 10^{-6}$ .

Algoritmul L-BFGS pentru diferire valori ale lui *m*, unde nx = 200 și ny = 200, cu parametrul Frank-Kamenetskii  $\lambda = 5$  furnizează o soluție cu caracteristicile ca în tabelul A5b.

<b>Tabelul</b> A <b>50.</b> I enformanța algoritmulul L-DI G5.									
$n = 40000$ . $\varepsilon = 10^{-6}$ .									
#iter #fg CPU									
L-BFGS (m=3)	721	763	80.08						
L-BFGS (m=5)	523	1308	58.06						
L-BFGS (m=7)	524	1846	58.61						
L-BFGS (m=9)	412	2272	47.10						

Tabelul A5b. Performanța algoritmului L-BFGS.

În tabelul A5c se prezintă performanțele metodei Newton trunchiate, unde nx = 200 și ny = 200, cu parametrul Frank-Kamenetskii  $\lambda = 5$ .

Tabelul A5c. Performanța algoritmului TN.

$\mathcal{E} = 10^{-6}$ .								
#iter #fg CPU								
<i>n</i> = 40000	27	477	52.82					

Figura A5 arată soluția problemei combustiei solidelor (problema variațională Bratu) pentru nx = 200 și ny = 200.



Fig. A5. Soluția aplicației A5. nx = 200, ny = 200. 40000 variabile.

Rezultatele obținute cu ACGHES sunt prezentate în tabelul A5d.

Table A5d	. Rezultatele	furnizate	de	ACGHES
-----------	---------------	-----------	----	--------

$\mathcal{E} = 10^{-6}$ .					
n	#iter	#fg	CPU (sec)	fx	
40000	299	333	28.20	-5.6114484936	

# 6. Configurația de energie minimă a unui grup de atomi

Această problemă, foarte dificilă, constă în a minimiza energia unei configurații de atomi sau molecule [Hoare, 1979]. Date pozițiile  $p_1, p_2, ..., p_n$  ale *n* atomi (puncte) în  $R^d$ , energia potențială este definită ca:

$$V_{d}(p) = \sum_{j=2}^{n} \sum_{i=1}^{j-1} v \Big( \big\| p_{j} - p_{i} \big\|_{2} \Big),$$

unde  $v: R \rightarrow R$  este funcția potențial între perechi de atomi. Funcția de potențial Lennard-Jones este definită ca [Lennard-Jones, 1931], [Morse, 1929]:

$$v(r) = r^{-12} - 2r^{-6}$$

Problema este de a determina o configurație (poziția celor *n* puncte) astfel încât funcția  $V_d$  să fie minimă [Northby, 1987]. Simularea interacțiunii între atomi este descrisă prin intermediul funcției de potențial interatomic v(r), care se presupune că depinde numai de distanța *r* dintre atomi. Funcția de energie satisface următoarele cerințe: este de două ori continuu diferențiabilă pe  $R_+$ , v(1) = -1,  $(r-1)\dot{v}(r) > 0$  pentru  $r \neq 1$ , și

$$\int_{1}^{\infty} r^2 |v(r)| \, \mathrm{d}r < \infty$$

Cele mai utilizate alegeri sunt potențialul Lennard-Jones și potențialul Morse. Potențialul Lennard-Jones descrie mai bine gazele nobile, în timp ce potențialul Morse descrie mai bine metalele. Tabelul A6a arată performanța algoritmilor de gradient conjugat pentru rezolvarea acestei probleme pentru n = 1000 atomi din  $R^3$  cu criteriul de performanță:  $\|g_k\|_{\infty} \le 10^{-6}$  sau  $\alpha_k |g_k^T d_k| \le 10^{-20} |f(x_{k+1})|$ .

Tabelul A6a. Performanța algoritmilor de gradient conjugat.

n-1	000	c - 1	$0^{-6}$
n = 1		– <u>२</u> = ।	U

	#iter	#fg	сри		#iter	#fg	сри
HS	1714	2207	279.43	LS-CD	2322	3182	403.15
FR	4001	5256	664.42	DL (t=1)	1851	2285	289.18
PRP	1987	2840	359.78	DL+(t=1)	1942	2516	318.20
PRP+	1942	1760	348.68	CONMIN	857	15001	1937.80
CD	3837	4498	569.32	CG_DESCENT	1896	3670	598.71
LS	2401	3446	435.28	SCG	1704	2198	278.07
DY	4001	4016	510.52	sPRP	2354	3352	423.52
TAS	2088	3033	383.06	sFR	4001	5256	664.62
PRP-FR	3394	4778	603.65	SCALCG	1252	2002	270.48
GN	1656	2451	309.51	ACGSD	1340	1654	209.30
HS-DY	1862	2431	307.43	aCGSD	1772	2391	302.28
hDY	1528	1907	242.28	PRP-SDC	2455	3149	399.37

Problema este grea, forma funcției v(r) implică o proastă condiționare. De fapt, valoarea funcției de minimizate este foarte aproape de epsilon mașină, ceea ce creează dificultăți tuturor algoritmilor de optimizare.

Tabelul A6b prezintă numărul de iterații, numărul de evaluări ale funcției de minimizat și timpul de calcul pentru rezolvarea acestei aplicații cu algoritmul L-BFGS pentru diferire valori ale lui *m*.

$n = 3000 \cdot \varepsilon = 10^{\circ}$ .					
	#iter	#fg	CPU		
L-BFGS (m=3)	1673	1755	269.04		
L-BFGS (m=5)	1944	3756	306.83		
L-BFGS (m=7)	1134	4950	183.33		
L-BFGS (m=9)	1097	6096	176.19		

**Tabelul A6b.** Performanța algoritmului L-BFGS.

Tabelul A6c prezintă numărul de iterații, numărul de evaluări ale funcției de minimizat și timpul de calcul (secunde) pentru rezolvarea acestei aplicații în care numărul de atomi este 1000 din spațiul  $R^3$ .

**Tabelul A6c.** Performanța algoritmului TN. #atomi=1000. d=3.  $\varepsilon = 10^{-6}$ .

	#iter	#fg	CPU
<i>n</i> = 3000	1523	64082	7379.42

Rezultatele obținute cu ACGHES sunt prezentate în tabelul A6d.

Table A6d. Rezultatele furnizate de ACGHES.

$\mathcal{E} = 10^{-6}$ .					
n	#iter	#fg	CPU (sec)	fx	
3000					

#### 7. Suprafețe minimale

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

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

unde  $f: K \to R$  este funcționala

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

și mulțimea K este definită prin:

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

pentru o anumită funcție  $v_D : \partial D \to R$ . Funcția  $v_D$  definește în mod unic soluția problemei suprafeței minimale [Nitsche, 1989]. O descriere a acestei probleme se găsește de asemenea în [Boyd și Vandenberghe, 2004, pg. 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

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

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  $z_{i,j} \in \mathbb{R}^2$  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:

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

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

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

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

În această formulare,  $f_{i,j}^L$  este definită numai pentru  $0 \le i \le n_x$  și  $0 \le j \le n_y$ , în timp ce  $f_{i,j}^U$  este definită pentru  $1 \le i \le n_x + 1$  și  $1 \le j \le n_y + 1$ .

Tabelul A7a arată performanța algoritmilor de gradient conjugat pentru rezolvarea acestei probleme pentru nx = 200 și ny = 200.

$hx = 200, hy = 200, \epsilon = 10$ .								
	#iter	#fg	сри			#iter	#fg	сри
HS	584	717	60.75		LS-CD	621	837	68.47
FR	1907	2010	174.15		DL (t=1)	489	554	47.23
PRP	646	917	74.31		DL+(t=1)	518	583	49.80
PRP+	771	1085	88.10		CONMIN	352	716	53.02
CD	819	918	78.12		CG_DESCENT	283	567	64.04
LS	486	677	55.05		SCG	514	594	50.53
DY	2252	2280	200.08		sPRP	618	870	70.76
TAS	669	938	76.13		sFR	1929	2038	176.93
PRP-FR	924	1275	104.78		SCALCG	348	466	34.93
GN	799	1118	90.72		ACGSD	451	475	41.28
HS-DY	652	816	67.91		aCGSD	589	638	55.11
hDY	652	816	67.90		PRP-SDC	463	499	43.15

**Tabelul A7a.** Performanța algoritmilor de gradient conjugat. mr = 200, mv = 200,  $c = 10^{-6}$ 

Algoritmul L-BFC	S furnizează	o solutie a acestei	aplicatii ca în	tabelul A7b.
0			·· · · · · · · · · · · · · · · · · · ·	

$n = 40000$ . $\varepsilon = 10^{-6}$ .						
#iter #fg CPU						
L-BFGS (m=3)	484	507	37.77			
L-BFGS (m=5)	459	977	35.93			
L-BFGS (m=7)	424	1408	33.94			
L-BFGS (m=9)	427	1844	35.10			

Tabelul A7b. Performanța algoritmului L-BFGS.

Algoritmul TN furnizează o soluție a acestei aplicații ca în tabelul A7c.

<b>Fabelul A7c.</b> Performanța algoritmului	TN.
$a - 10^{-6}$	

$\mathcal{E} = 10^{-6}$ .					
	#iter	#fg	CPU		
n = 40000	23	327	26.63		

O reprezentare a suprafeței cu arie minimă în aceste condiții pe frontieră este ilustrată în figura A7a.



Fig. A7a. Soluția aplicației. nx = 200, ny = 200. 40000 variabile.

Rezultatele obținute cu ACGHES sunt prezentate în tabelul A7d.

	$\mathcal{E} = 10^{-6}$ .						
	n	#iter	#fg	CPU (sec)	fx		
	40000	281	308	16.24	1.0		
0							

Soluția furnizată de ACGHES este ilustrată în figura A7b.



Fig. A7b. Soluția aplicației dată de ACGHES. nx = 200, ny = 200. 40000 variabile.

În tabelul A8, pentru fiecare prototip, se prezintă algoritmii care ocupă primele două poziții din punctul de vedere al timpului de calcul.

Primele două poziții. Timpul de calcul. $\varepsilon = 10^{\circ}$ .						
Prototipul	Poziția 1	Poziția 2				
1	CONMIN	SCALCG				
2	SCALCG	DY				
3	ACGSD	PRP-SDC				
4	CONMIN	hDY				
5	ACGSD	SCALCG hDY				
6	ACGSD					
7	SCALCG	ACGSD				

Tabelul A8. Cei mai buni algoritmi de gradient conjugat.

În tabelul A9 se arată o comparație între algoritmii de gradient conjugat și algoritmul L-BFGS, pentru rezolvarea acestor aplicații, din punctul de vedere al timpului de calcul. În această comparație s-au considerat cele mai bune rezultate obținute de acești algoritmi în ceea ce privește timpul de calcul, resursa cea mai importantă.

	Gradientul Conjugat			L-BFGS		
Prototipul	#iter	#iter #fg CPU		#iter	#fg	CPU
1	242	486	30.04	322	338	20.90
2	876	1143	72.19	788	1750	52.15
3	1098	1137	108.22	659	3871	57.93
4	7419	15001	69.36	1904	2001	9.45
5	705	731	87.62	412	2272	47.10
6	1340	1654	209.30	1097	6096	176.19
7	348	466	34.93	424	1408	33.94

**Tabelul A9.** Comparație între algoritmii de gradient conjugat și L-BFGS. Timpul de calcul (secunde).

După cum se vede, algoritmul L-BFGS este superior tuturor algoritmilor de gradient conjugat. Algoritmul L-BFGS în implementarea lui Nocedal [1980], în care căutarea liniară se face prin intermediul condițiilor Wolfe în implementarea Moré-Thuente, în esența lui este mult diferit de algoritmii de gradient conjugat, atât în ceea ce privește calculul direcției de deplasare, cât și a lungimii pasului. Se pare că algoritmul L-BFGS calculează o direcție mai profitabilă în comparație cu algoritmii de gradient conjugat.

**8.** Problema inversă în asimilarea datelor în meteorologie [Derber, 1987], [Le Dimet și Talagrand, 1986], [Long, Thacker, 1989], [Zou, *et all*, 1993], [Das, Meirovitch, Navon, 2003], [Daescu, Navon, 2007].

Un obstacol major în previziunea vremii – una dintre cele mai importante probleme ale societății contemporane – este determinarea stării inițiale a atmosferei care să ne permită rezolvarea sistemelor de ecuații asociate acestui sistem fizic. Deși dispunem de o cantitate impresionantă de date furnizată de o multitudine de sateliți geostaționari, totuși aceste date sunt insuficiente. Aceasta reclamă dezvoltarea unor metode numerice care să extragă maximul de informații din aceste date pentru a completa descrierea sistemului (atmosferic, oceanic, etc.). Una dintre aceste metode numerice este asimilarea variațională patru-dimensională. În esență aceasta este o problemă de cele mai mici pătrate în care funcția obiectiv constă în a minimiza distanța dintre câmpul predicționat de model și câmpul

măsurat experimental, în funcție de un set de condiții inițiale, în virtutea unor restricții asupra variabilelor date de ecuațiile de mișcare. Presupunem că dispunem de un set de vectori de observații  $z_i$  corespunzători momentelor de timp  $t_i$ ,  $1 \le i \le N$ . Fiecare vector de observații constă dintr-o mulțime de date meteorologice privind: temperatura, umiditatea, viteza și tăria vântului, radianța etc. Presupunem că avem un model matematic definit de un set de ecuații neliniare cu derivate parțiale – în mod normal ecuațiile Navier-Stokes – care sunt discretizate atât în spațiu cât și în timp, ceea ce conduce la un sistem de ecuații algebrice neliniare, pe care le putem scrie sub forma:

$$x_{i+1} = x_i + \Delta t G_i(x_i),$$

unde  $x_i \in \mathbb{R}^n$  este un vector care descrie starea modelului la momentul de timp  $t_i$  și operatorul  $G_i$  depinde de schema de discretizare. Precizarea condițiilor inițiale  $x_0$  la momentul de timp  $t_0$  determină o soluție unică a modelului.

Compararea datelor obținute experimental (prin măsurători) cu cele furnizate de model nu este simplă. Într-adevăr, este posibil ca parametrii observați experimental să fie diferiți în natură de cei descriși de model, de exemplu modelul conține temperaturi, dar datele experimentale se pot referi la cantitatea de radiații. De asemenea, în modelul discretizat observațiile se pot face în puncte care nu aparțin rețelei de discretizare. Deci este necesar să transformăm variabilele  $x_i$  ale modelului în sensul de a le compatibiliza cu observațiile. Să presupunem că aceasta se face prin intermediul transformării  $\hat{x}_i = h_i(x_i)$ , unde  $h_i$  este operatorul de observare asociat vectorului de observații  $z_i$ .

Cu acestea funcția obiectiv se definește ca distanța dintre câmpul predicționat de model  $x_i$  și vectorul de observații  $z_i$ :

$$J = \frac{1}{2} \sum_{i=1}^{N} (h_i(x_i) - z_i)^T W(h_i(x_i) - z_i).$$

Ideal matricea de ponderare W este inversa matricei erorii de covarianță, dar în practică aceasta este o matrice diagonală care aproximează această informație. Problema de optimizare constă deci în a *determina vectorul condițiilor inițiale*  $x_0$  *astfel încât șirul de stări*  $x_1,...,x_N$  *date de model minimizează funcția de distanță* J. Este foarte important să vedem structura și mărimea unei astfel de probleme de optimizare. În aplicațiile curente numărul de variabile n este de ordinul  $10^6$ . Pe de altă parte, numărul de intervale de timp N este de ordinul  $10^2$ . Evaluarea funcției obiectiv este foarte costisitoare deoarece implică rezolvarea modelului asociat evoluției atmosferei. Mai mult, o problemă crucială este aceea calculului gradientului funcției de minimizat. O tehnică pentru calculul gradientului  $\nabla J$  constă în utilizarea modelului adjunct. O altă dificultate în rezolvarea acestor probleme o constituie faptul că transformările  $h_i$  fac Hessianul funcției J o matrice densă.

Ecuațiile modelului care descrie mișcarea fluidului la joasă altitudine sunt următoarele:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} - fv + \frac{\partial \phi}{\partial x} = 0,$$
  
$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + fu + \frac{\partial \phi}{\partial y} = 0,$$
  
$$\frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial x} + v \frac{\partial \phi}{\partial y} + \phi \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) = 0,$$

cu condiții la limită periodice pe direcția est-vest și condiții fixe pe direcția nord-sud. În acest model u, v și  $\phi$  sunt cele două componente ale câmpului de viteze și respectiv câmpul geopotențial. Aceste variabile sunt discretizate în spațiu și timp utilizând o schemă de diferențe centrale. Experimentele numerice sunt efectuate într-un domeniu dreptunghiular cu mărimea L=6000 Km și D=4400 Km [Zou, Navon și Le Dimet, 1992].

Sistemul adjunct corespunzător ecuațiilor de mai sus se bazează pe aproximarea liniară a acestor ecuații:

$$\frac{\partial u'}{\partial t} + \frac{\partial (uu')}{\partial x} + v' \frac{\partial u}{\partial y} + v \frac{\partial u'}{\partial y} - fv' + \frac{\partial \phi'}{\partial x} = 0,$$
  
$$\frac{\partial v'}{\partial t} + \frac{\partial (vv')}{\partial y} + u' \frac{\partial v}{\partial x} + u \frac{\partial v'}{\partial x} + fu' + \frac{\partial \phi'}{\partial y} = 0,$$
  
$$\frac{\partial \phi'}{\partial t} + \frac{\partial (\phi'u)}{\partial x} + \frac{\partial (\phi u')}{\partial x} + \frac{\partial (\phi v')}{\partial y} + \frac{\partial (\phi'v)}{\partial y} = 0,$$

care în formă matriceală se exprimă ca

$$\frac{\partial X}{\partial t} + AX = 0,\tag{1}$$

unde vectorul  $X = [u' v' \phi']$  și

$$A = \begin{bmatrix} \frac{\partial(u(.))}{\partial x} + v \frac{\partial((.))}{\partial y} & (.) \frac{\partial u}{\partial y} - f & \frac{\partial(.)}{\partial x} \\ (.) \frac{\partial v}{\partial x} + f & u \frac{\partial(.)}{\partial x} + \frac{\partial(v(.))}{\partial y} & \frac{\partial(.)}{\partial y} \\ \frac{\partial(\phi(.))}{\partial x} & \frac{\partial(\phi(.))}{\partial y} & \frac{\partial(u(.))}{\partial x} + \frac{\partial(v(.))}{\partial y} \end{bmatrix}$$

Vectorul X reprezintă o perturbație în jurul stării  $u, v, \phi$ . Cu acestea, sistemul adjunct al sistemului (1) este următorul:

$$-\frac{\partial \hat{X}}{\partial t} + A^* \hat{X} = 0.$$
<sup>(2)</sup>

Operatorul adjunct  $A^*$  satisface relația de definiție:

$$\left\langle AX, \hat{X} \right\rangle_{D \times T} = \left\langle A^* \hat{X}, X \right\rangle_{D \times T},$$

unde  $\langle . \rangle$  reprezintă produsul scalar definit ca

$$\langle X, Y \rangle_{D \times T} = \sum_{i=1}^{3} \int_{t_0}^{t_R} \int_{D} X_i Y_i dD dt,$$

unde T este domeniul timpului în care  $t_0$  și  $t_R$  sunt capetele acestui interval de timp, iar D este domeniul spațial peste care aceste ecuații sunt integrate. Le Dimet [1988] definește operatorul adjunct  $A^*$  sub forma:

$$A^{*} = \begin{bmatrix} -u\frac{\partial(.)}{\partial x} - \frac{\partial(v(.))}{\partial y} & (.)\frac{\partial v}{\partial x} + f & -\phi\frac{\partial(.)}{\partial x} \\ (.)\frac{\partial u}{\partial y} - f & -v\frac{\partial(.)}{\partial y} - \frac{\partial(u(.))}{\partial x} & -\phi\frac{\partial(.)}{\partial y} \\ -\frac{\partial(.)}{\partial x} & -\frac{\partial(.)}{\partial y} & -u\frac{\partial(.)}{\partial x} - v\frac{\partial(.)}{\partial y} \end{bmatrix}.$$

Cu acestea funcția obiectiv de minimizat se definește ca o sumă ponderată a pătratelor diferenței dintre observații și valorile determinate de model:

$$J(X_0) = W(X(X_0) - X_{obs})^T (X(X_0) - X_{obs})$$
  
=  $W_{\phi} \sum_{j=1}^{N_{\phi}} (\phi_j - \phi_{obs})^2 + W_V \sum_{j=1}^{N_V} \left[ (u_j - u_{obs})^2 + (v_j - v_{obs})^2 \right],$ 

unde  $N_{\phi}$  este numărul total de observații geopotențiale disponibile în intervalul de timp  $[t_0, t_R]$  și  $N_V$  este numărul total de observații ale vectorului vitezei vântului. Elementele  $u_{obs}$ ,  $v_{obs}$  și  $\phi_{obs}$  sunt valorile observate ale stării sistemului, aici componenta în direcția nord a vântului, componenta în direcția est a vântului și respectiv câmpul geopotențial. Elementele u, v și  $\phi$  sunt valorile furnizate de model, adică obținute prin integrarea modelului. Factorii de ponderare  $W_{\phi}$  și  $W_V$  sunt inversa estimațiilor erorilor medii ale câmpului geopotențial și respectiv a componentelor vântului. În experimentele numerice s-a considerat  $W_{\phi} = 10^{-4} m^{-2} s^2$  și  $W_V = 10^{-2} m^{-2} s^2$ .

Gradientul funcției obiectiv în raport cu condițiile inițiale  $X(0) = X_0$  este  $\nabla_{X_0} J = \hat{X}(0)$ , unde  $\hat{X}(0)$  este obținut prin integrarea sistemului adjunct (2) înapoi în timp cu condiții inițiale nule la timpul  $t_R$ .

Pentru discretizarea  $\Delta x = 300 Km$ ,  $\Delta y = 220 Km$  și  $\Delta t = 600 s$ , problema în varianta discretizată conține n = 14763 variabile. Evoluția funcției de minimizat furnizată de algoritmul LBFGS în implementarea dată de Liu și Nocedal este ilustrată în figura A8.

Tabelul A10 arată performanța algoritmului Newton Trunchiat în implementarea TN a lui Nash, iar figura A9 prezintă evoluția funcției de minimizate de-a lungul celor 56 de iterații.

Tabelul A10. Performanța algoritmului TN.

	#iter	#fg	CPU	
<i>n</i> = 14763	56	204	10.02	



Fig. A8. Evoluția funcției de minimizat pentru diferite valori ale lui m.



Fig. A9. Evoluția funcției de minimizat pentru problema inversă în asimilarea datelor în meteorologie

În tabelul A11 prezentăm o comparație între algoritmii L-BFGS și Newton trunchiat în implementările date de Nocedal [1980] și respectiv Nash [1985], pentru rezolvarea acestor aplicații, din punctul de vedere al timpului de calcul.

	L-BFGS			TN		
Prototipul	#iter	#iter #fg CPU		#iter	#fg	CPU
1	322	338	20.90	13	307	19.75
2	788	1750	52.15	35	777	50.25
3	659	3871	57.93	56	1817	161.77
4	1904	2001	9.45	350	6581	33.54
5	412	2272	47.10	27	477	52.82
6	1097	6096	176.19	1523	64082	7379.42
7	424	1408	33.94	23	327	26.63
8	61	129	4.53	56	204	10.02

**Tabelul A11.** Comparație între algoritmii L-BFGS și TN

Câteva concluzii se impun cu necesitate.

1) Studiile numerice prezentate în acest capitol cât și aplicațiile practice arată că algoritmii de gradient conjugat sunt capabili să rezolve eficient probleme complexe de optimizare fără restricții cu număr mare de variabile.

2) Algoritmii sunt extrem de simpli și se pot imediat implementa în programe de calcul care utilizează câțiva vectori. Pentru probleme cu un număr mare de variabile ( $\geq 10^7$ ) implementările pot beneficia de tehnica de programare "loop unrolling" de diferite adâncimi, cum este cazul pachetului CG\_DESCENT.

3) După cum vedem algoritmii de gradient conjugat scalat SCALCG și CONMIN se dovedesc a fi cei mai performanți în această clasă. În imediata vecinătate se află CG DESCENT.

4) Algoritmii quasi-Newton cu memorie limitată se dovedesc a fi cei mai performanți. Eficiența acestora derivă din faptul că aceștia țin seama de curbura funcției de minimizat, ceea ce algoritmii de gradient conjugat nu realizează.

5) Ca o concluzie a studiului numeric de mai sus, precum și a prototipurilor rezolvate și prezentate în această secțiune putem spune că algoritmul Newton trunchiat în implementarea TN a lui Nash [1985]<sup>4</sup> este depășit atât de algoritmii de gradient conjugat cu precondiționare SCALCG [Andrei, 2007a,b,c,d,e]<sup>5</sup> și CONMIN [Shanno și Phua, 1976, 1980]<sup>6</sup>, cât și de algoritmii quasi Newton BFGS cu memorie limitată [Liu și Nocedal, 1989]<sup>7</sup>,

<sup>&</sup>lt;sup>4</sup> Nash, S.G., (1985) *Preconditioning of truncated-Newton methods*. SIAM J. on Scientific and Statistical Computing, 6 (1985), pp.599-616.

<sup>&</sup>lt;sup>5</sup> Andrei, N., (2007a) Scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization. Optimization Methods and Software, 22 (2007), pp.561-571.

Andrei, N., (2007b) A scaled BFGS preconditioned conjugate gradient algorithm for unconstrained optimization. Applied Mathematics Letters, 20 (2007), pp.645-650.

Andrei, N., (2007c) *Scaled conjugate gradient algorithms for unconstrained optimization*. Computational Optimization and Applications. In press.

**Andrei**, N., (2007d) *A scaled nonlinear conjugate gradient algorithm for unconstrained optimization*. Optimization. A journal of mathematical programming and operations research, In press.

Andrei, N., (2007e) A Dai-Yuan conjugate gradient algorithm with sufficient descent and conjugacy conditions for unconstrained optimization. Applied Mathematics Letters. In press.

<sup>&</sup>lt;sup>6</sup> Shanno, D.F., Phua, K.H., (1976) Algorithm 500, Minimization of unconstrained multivariate functions, ACM Trans. on Math. Software, 2 (1976) 87-94.

Shanno, D.F., Phua, K.H., (1980) *Remark on algorithm 500*. ACM Trans. on Math. Software, 6 (1980), pp. 618-622.

<sup>&</sup>lt;sup>7</sup> Liu, D.C., Nocedal, J., (1989) On the limited memory BFGS method for large scale optimization. Mathematical Programming, vol.45, 1989, pp.503-528.

[Nocedal, 1980]<sup>8</sup>. Observăm că în algoritmul TN este vorba de o dublă aproximare. În primul rând se aproximează matricea Hessian prin actualizarea BFGS, apoi sistemul Newton este rezolvat în mod aproximativ conform filosofiei metodelor Newton trunchiate. Aceasta face ca rata de convergență a metodei Newton trunchiate să fie inferioară ratei de convergență a metodelor de gradient conjugat sau quasi-Newton BFGS cu memorie limitată. Cu toate acestea, metoda Newton trunchiată rămâne una dintre metodele de bază pentru rezolvarea problemelor de optimizare fără restricții.

În continuare considerăm 5 prototipuri (A1, A2, A3, A5 și A7) pentru care pasul de discretizare este nx = ny = 400,500,1000. Aceasta ne conduce la probleme cu 160000, 250000 si respectiv 1000000 de variabile. Tabelul A12 prezintă performanțele algoritmului SCALCG în ceea ce privește timpul de calcul pentru rezolvarea acestor aplicații de mari dimensiuni [Andrei, 2007f]<sup>9</sup>.

_	$\theta$ spectral, criteriul Powell, $\ \nabla f(x_k)\ _{\infty} \leq 10^{-6}$ .							
		Timpul de calcul cpu (secunde)						
	п	A1 A2		A3	A5	A7		
-	160000	58.19	197.14	532.25	246.95	113.77		
	250000	97.88	361.60	929.44	370.85	250.43		
	1000000	744.11	2938.01	5169.71	3027.08	1831.83		

Tabelul A12. Performanta algoritmului SCALCG.

Figura A10 arată evoluția timpului de calcul cpu în funcție de numărul de variabile.



Fig.A10. Evoluția timpului de calcul cpu.

În general, putem spune că proprietățile de convergență ale acestor algoritmi sunt bine cunoscute și ca o ironie a dezvoltărilor teoretice, algoritmii pentru care se pot demonstra teoreme de convergență au o comportare numerică mai modestă fată de cei pentru care nu s-a demonstrat convergența. Rata

<sup>&</sup>lt;sup>8</sup> Nocedal, J., (1980) Updating quasi-Newton matrices with limited starage. Mathematics of Computation, 35 (1980), pp.773-782.

Andrei, N., (2007f) Large-scale unconstrained optimization Minpack-2 applications solved with SCALCG. În Works 2007, Manuscris, Biblioteca Academiei Române, September 4, 2007, pg.1-5.

de convergența este superliniară și lungimea pasului de deplasare nu este unitară ca în cazul algoritmilor de tip Newton.

#### **Bibliografie**

- R., Aris, The mathematical theory of diffusion and reaction in permeable catalysts. Oxford, 1975.
- **B.M.**, Averick, Carter, R.G., Moré, J.J., *The MINPACK-2 test problem collection (Preliminary version)*, Argonne National Laboratory, Thechnical Memorandum No. 150, May 1991.
- J., Bebernes, Eberly, D., *Mathematical problems from combustion theory*. Applied Mathematical Sciences 83, Springer-Verlag, 1989.
- S., Boyd, Vandenberghe, L., Convex Optimization. Cambridge University Press, Cambridge, 2004.
- **G.**, **Cimatti, Menchi, O**., (1978) On the numerical solution of a variational inequality connected with the hydrodynamic lubrication of a complete journal bearing. Calcolo, 15, (1978), pp.249-258.
- **D.N., Daescu, Navon, I.M.**, Efficiency of a POD-based reduced second-order adjoint model in 4D-Var data assimilation. Int. J. for Numer. Meth. Fluids, 53 (2007), pp.985-1004.
- B., Das, Meirovitch, H., Navon, I.M., Performance of hybrid methods for large-scale unconstrained optimization as applied to models of proteins. J. Comput. Chem., 24 (2003), pp.1222-1231.
- J.C., Derber, Variational four-dimensional analysis using quasi-geostrophic constraints. Mon. Wea. Review, 115 (1987), pp.998-1008.
- **C.M., Elliot, Ockendon, J.R.**, *Weak and variational methods for moving boundary problems*. Research Notes in Mathematics, vol.59, Pittman, 1982.
- J., Garner, Benedek, R. Solution of Ginzburg-Landau equations for inhomogeneous superconductors by nonlinear optimization. Phys. Rev., B. 42, 1990, pp.376-385.
- R., Glowinski, Numerical Methods for Nonlinear Variational Problems. Springer-Verlag, Berlin, 1984.
- J., Goodman, Kohn, R., Reyna, L., Numerical study of a relaxed variational problem from optimal design. Comput. Methods Appl. Mech. Engrg., 57, 1986, pp.107-127.
- M.R., Hoare, Structure and dynamics of simple microclusters. Advances in Chemical Physics, 40, 1979, pp.49-135.
- Le Dimet, F.X., (1988) Determination of the adjoint of a numerical weather prediction model. Technical Report FSU-SCRIE-88-79, Florida State University, Tallahassee, Florida 32306-4052, 1988, 22 pp.
- F.X., Le Dimet, Talagrand, O., Variational algorithms for analysis and assimilation of meteorological observations: Theoretical aspects. Tellus, 38A (1986), pp.97-110.
- Y., Lin, 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.
- **R.B., Long, Thacker, W.C.,** *Data assimilation into a numerical equatorial ocean model. Part 2: Assimilation experiments.* Dyn. Atmos. Oceans, 13 (1989), pp.465-477.
- S.G., Nash, (1985) *Preconditioning of truncated-Newton methods*. SIAM J. on Scientific and Statistical Computing 6, 599-616 (1985).
- J.C.C., Nitsche, Lectures on minimal surfaces. Vol.1, Cambridge University Press, 1989.
- **J.**, **Nocedal**, *Updating quasi-Newton matrices with limited starage*. Mathematics of Computation, 35 (1980), pp.773-782.
- J.A., Northby, Structure and binding of Lennard-Jones clusters: 13<n<147. Journal of Chemical Physics, 87, 1987, pp.6166-6177.
- **D.P., O'Leary, Yang, W.H.**, *Elastoplastic torsion by quadratic programming*, Comput. Methods Appl. Mech. Engrg., 16, 1978, pp.361-368.
- Zou, X., Navon, I.M., Berger, M., Phua, P.K.H., Schlick, T., Le Dimet, F.X., (1993) Numerical experience with limited-memory and truncated Newton methods. SIAM J. Opt., 3 (1993), pp.582-608.