

# Rezolvarea unor aplicații din colecția MINPACK-2 utilizând algoritmi de gradient conjugat

Neculai Andrei

*Research Institute for Informatics,  
Center for Advanced Modeling and Optimization,  
8-10, Averescu Avenue, Bucharest 1, Romania  
E-mail: nandrei@ici.ro*

**Rezumat.** Lucrarea prezintă performanța algoritmilor de gradient conjugat în ceea ce privește rezolvarea problemelor de minimizare din colecția MINPACK-2. În acest sens se consideră un număr de șase aplicații de minimizare din MINPACK-2 care sunt rezolvare cu 20 de algoritmi de gradient conjugat implementați în pachetul CGALL. Fiecare aplicație este descrisă pe scurt împreună cu transformarea ei într-o problemă de optimizare.

MSC: 49M07, 49M10, 90C06, 65K

*Cuvinte cheie:* metode de gradient conjugat, comparații numerice, aplicații de programare matematică

## 1. Introducere

În această lucrare prezentăm experiența numerică privind rezolvarea unor aplicații industriale concrete din diverse domenii de interes atât pentru știință cât și pentru practica industrială. În acest sens ne vor referi la colecția MINPACK-2 asamblată de Jorge J. Moré și colaboratorii lui de la Argonne National Laboratory, Argonne, Illinois, USA. Într-adevăr, Army High Performance Computing Research Center at the University of Minnesota și The Mathematics and Computer Science Division at Argonne National Laboratory sunt într-o permanentă colaborare asupra dezvoltării unui toolbox pentru optimizare fără restricții - MINPACK-2. O componentă importantă a acestui proiect o constituie activitatea de construcție a unei colecții de probleme de optimizare din domeniul științific și industrial care să constituie probleme de test pentru pachetul MINPACK-2.

Având în vedere această colecție de probleme de optimizare fără restricții, prezentată foarte succint în câteva rapoarte de cercetare, precum și sub forma unor subrutine Fortran care implementează modelele matematice ale acestor probleme, în această fază a proiectului GRIDMOSI ne punem problema prezentării rezultatelor computaționale ale algoritmilor noștri, precum și prezentarea unei comparații între algoritmii considerați în acest studiu. Problemele considerate sunt următoarele:

- 1) Elastic-Plastic Torsion Problem,
- 2) Pressure Distribution in a Journal Bearing Problem,
- 3) Optimal Design with Composite Materials Problem,
- 4) Inhomogeneous Superconductors. 1-dimensional Ginzburg-Landau Problem,
- 5) Steady State Combustion. Solid Fuel Ignition.
- 6) Lennard-Jones Cluster Problem,

Prezentarea acestor probleme se găsește în: Brett M. Averick, Richard G. Carter and Jorge J. Moré, *The MINPACK-2 test problem collection (Preliminary version)*, Mathematics and Computer Science Division, Argonne National Laboratory, Technical Memorandum No. 150, May 1991.

În cele ce urmează vom detalia pe scurt modelele matematice ale acestor probleme. Cititorul interesat găsește detalii în acest raport tehnic, precizat mai sus, precum și în lucrările menționate la bibliografia lucrării.

### **Aplicația A1. “Elastic-Plastic Torsion Problem”**

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:

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

unde  $\text{dist}(\cdot, \partial D)$  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:

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

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 lățe 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

$$q_{i,j}^L(v) = \mu_{i,j} \left\{ \left( \frac{v_{i+1,j} - v_{i,j}}{h_x} \right)^2 + \left( \frac{v_{i,j+1} - v_{i,j}}{h_y} \right)^2 \right\},$$
$$\mu_{i,j} = \frac{h_x h_y}{6} \{ w_q(z_{i,j}) + w_q(z_{i+1,j}) + w_q(z_{i,j+1}) \}.$$

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

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

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

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  $z_{i,j}$ .

Considerând  $D = (0,1) \times (0,1)$ ,  $c = 5$  și  $n_x = 200$ ,  $n_y = 200$ , atunci performanța algoritmilor de gradient conjugat implementați în CGALL este prezentată în Tabelul A1.

**Tabelul A1.** Performanța lui CGALL. stoptest=1. angle restart.  
nx = 200, ny = 200.

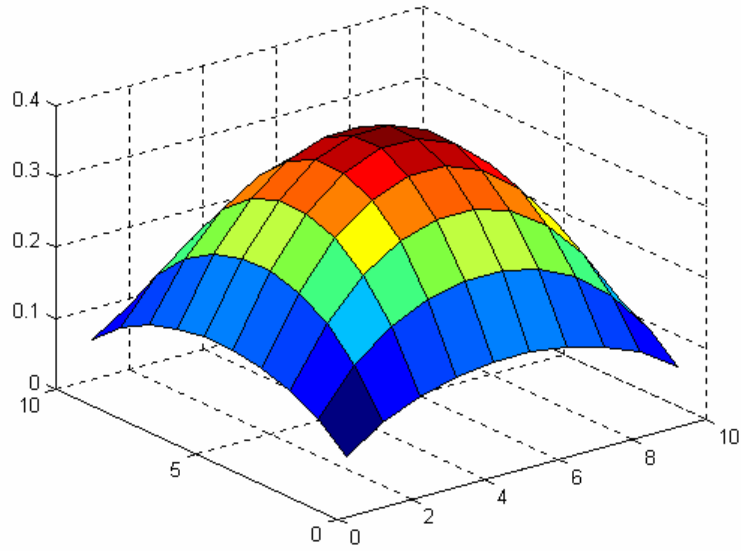
CG with angle restart. Oct. 6, 2006							
1	CG-HS Algorithm:Elastic-Plastic Torsion Problem Hestenes and Stiefel. stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	470	0	606	135	4494	-.4392669723172E+00	.5472010712090E-04
TOTAL	470	0	606	135	44.94 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
2	CG-FR Algorithm:Elastic-Plastic Torsion Problem Fletcher and Reeves. stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	1123	0	1211	87	9390	-.4392678170513E+00	.5542879589180E-04
TOTAL	1123	0	1211	87	93.90 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
3	CG-PRP Algorithm:Elastic-Plastic Torsion Problem Polak-Ribiere and Polyak. stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	549	0	790	240	5672	-.4392660458110E+00	.5899288073497E-04
TOTAL	549	0	790	240	56.72 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
4	CG-PRP+ Algorithm:Elastic-Plastic Torsion Problem Polak-Ribiere and Polyak plus. stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	515	0	744	228	5330	-.4392675484265E+00	.4671741916588E-04
TOTAL	515	0	744	228	53.30 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							

5	CG-CD Algorithm:Elastic-Plastic Torsion Problem Conjugate Descent - Fletcher. stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	772	1	851	78	6528	-.4392677730748E+00	.4964667426406E-04
TOTAL	772	1	851	78	65.28 (seconds)	proc=	.13%
CG with angle restart. Oct. 6, 2006							
6	CG-LS Algorithm:Elastic-Plastic Torsion Problem Liu and Storey. stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	419	0	563	143	4098	-.4392676304000E+00	.5505354551139E-04
TOTAL	419	0	563	143	40.98 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
7	CG-DY Algorithm:Elastic-Plastic Torsion Problem Dai and Yuan. stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	464	0	489	24	3794	-.4392678118671E+00	.4636380569893E-04
TOTAL	464	0	489	24	<b>37.94 (seconds)</b>	proc=	.00%
CG with angle restart. Oct. 6, 2006							
8	CG-DL Algorithm:Elastic-Plastic Torsion Problem Dai and Liao (t=1). stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	474	0	547	72	4166	-.4392676276893E+00	.5899636211465E-04
TOTAL	474	0	547	72	41.66 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
9	CG-DL+ Algorithm:Elastic-Plastic Torsion Problem Dai and Liao plus (t=1). stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	438	0	498	59	3796	-.4392675732838E+00	.6215678629793E-04
TOTAL	438	0	498	59	37.96 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
10	CG-SDC Algorithm:Elastic-Plastic Torsion Problem Andrei - Sufficient Descent Condition. stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	553	0	628	74	4804	-.4392677539163E+00	.5386515413574E-04
TOTAL	553	0	628	74	48.04 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
11	CG-hDY Algorithm:Elastic-Plastic Torsion Problem Hybrid Dai and Yuan. max(c*DY,min(HS,DY)). stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	480	0	626	145	4593	-.4392670489175E+00	.5213096087434E-04
TOTAL	480	0	626	145	45.93 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
12	CG-hDYz Algorithm:Elastic-Plastic Torsion Problem Hybrid Dai and Yuan. max(0,min(HS,DY)). stoptest= 1						Function

n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	525	0	684	158	5026	-.4392673743621E+00	.5806031275899E-04
TOTAL	525	0	684	158	50.26 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
13	CG-GN Algorithm:Elastic-Plastic Torsion Problem Gilbert and Nocedal. stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	451	0	646	194	4627	-.4392665762755E+00	.6168428335805E-04
TOTAL	451	0	646	194	46.27 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
14	CG-HuS Algorithm:Elastic-Plastic Torsion Problem Hu and Storey. max(0,min(PRP,FR)). stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	592	0	829	236	5989	-.4392665893852E+00	.7078891172025E-04
TOTAL	592	0	829	236	59.89 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
15	CG-TaS Algorithm:Elastic-Plastic Torsion Problem Touat-Ahmed and Storey. (PRP <= FR). stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	619	0	877	257	6333	-.4392675027509E+00	.5154553939208E-04
TOTAL	619	0	877	257	63.33 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
16	CG-LS&CD Algorithm:Elastic-Plastic Torsion Problem Hybrid Liu&Storey-Conjugate Descent. max(0,min(LS,CD)) stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	528	0	734	205	5307	-.4392672120214E+00	.5198145401967E-04
TOTAL	528	0	734	205	53.07 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
17	CG-BM Algorithm:Elastic-Plastic Torsion Problem Birgin and Martinez (scaled Perry). Theta spectral. stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	539	0	663	123	4978	-.4392676103086E+00	.6174792411862E-04
TOTAL	539	0	663	123	49.78 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
18	CG-BM+ Algorithm:Elastic-Plastic Torsion Problem Birgin and Martinez plus (scaled Perry). Theta spectral. stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	512	0	633	120	4745	-.4392675527216E+00	.5784243301837E-04
TOTAL	512	0	633	120	47.45 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
19	CG-sPRP Algorithm:Elastic-Plastic Torsion Problem Scaled Polak-Ribiere and Polyak. Theta spectral. stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	507	0	728	220	5244	-.4392661393681E+00	.6507907916066E-04

TOTAL	507	0	728	220	52.44 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
20	CG-sFR Algorithm:Elastic-Plastic Torsion Problem					Function	
	Scaled Fletcher and Reeves. Theta spectral. stoptest= 1						
n	iter	irs	fgcnt	lsent	time(c)	fxnew	gnorm
40000	1107	0	1194	86	9328	-.4392678165961E+00	.6063685566256E-04
TOTAL	1107	0	1194	86	93.28 (seconds)	proc=	.00%

Rezultate numerice privind această problemă sunt prezentate de exemplu în Elliot și Ockendon [1982], O’Leary și Yang [1989] și Moré și Toraldo [1991]. Pentru  $nx = 10$ ,  $ny = 10$  soluția problemei este ilustrată în Figura A1.



**Fig. A1.** Soluția problemei “Elastic-Plastic Torsion“  $nx = 10$ ,  $ny = 10$ .

#### **Aplicația A2. “Pressure Distribution in a Journal Bearing Problem“**

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 asta  $w_q(\xi_1, \xi_2) = (1 + \varepsilon \cos \xi_1)^3$  și  $w_l(\xi_1, \xi_2) = \varepsilon \sin \xi_1$ . Domeniul de admisibilitate este mulțimea

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

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  $n_x=200$  și  $n_y=200$ , atunci performanța algoritmilor incluși în CGALL este ca în Tabelul A2.

**Tabelul A2.** Performanța lui CGALL. stoptest=1. angle restart.  
nx = 200, ny = 200.

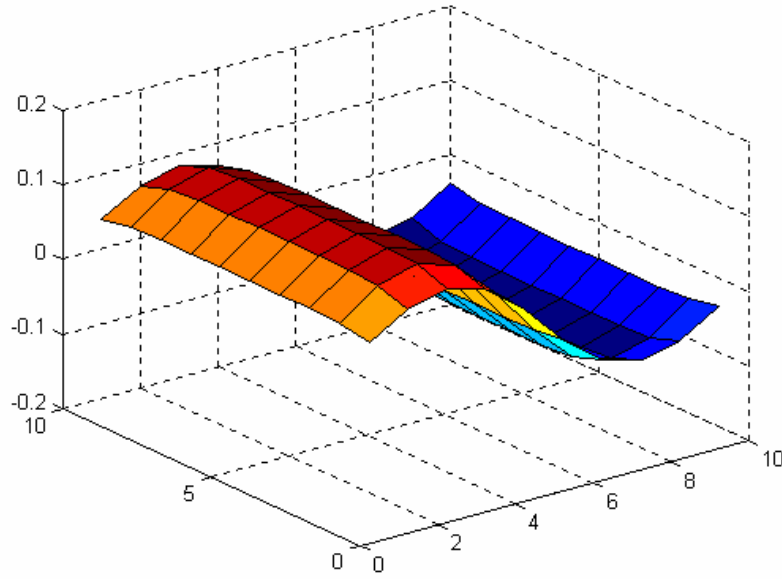
CG with angle restart. Oct. 6, 2006							
1	CG-HS Algorithm:Pressure Distribution Problem Hestenes and Stiefel. stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	1373	0	1769	395	13475	-.2828928441305E+00	.2242178913041E-04
TOTAL	1373	0	1769	395	134.75 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
2	CG-FR Algorithm:Pressure Distribution Problem Fletcher and Reeves. stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	2001	0	2114	113	16813	-.2828928829736E+00	.1830398417199E-03
TOTAL	2001	0	2114	113	168.13 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
3	CG-PRP Algorithm:Pressure Distribution Problem Polak-Ribiere and Polyak. stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	1242	0	1773	530	13045	-.2828927665800E+00	.3781489191363E-04
TOTAL	1242	0	1773	530	130.45 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
4	CG-PRP+ Algorithm:Pressure Distribution Problem Polak-Ribiere and Polyak plus. stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	1328	0	1891	562	13945	-.2828928567816E+00	.3004022749413E-04
TOTAL	1328	0	1891	562	139.45 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
5	CG-CD Algorithm:Pressure Distribution Problem Conjugate Descent - Fletcher. stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	1884	3	2112	227	16466	-.2828929397215E+00	.2709597886241E-04
TOTAL	1884	3	2112	227	164.66 (seconds)	proc=	.16%
CG with angle restart. Oct. 6, 2006							
6	CG-LS Algorithm:Pressure Distribution Problem Liu and Storey. stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	938	0	1333	394	9756	-.2828927981745E+00	.2494279574639E-04
TOTAL	938	0	1333	394	97.56 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
7	CG-DY Algorithm:Pressure Distribution Problem Dai and Yuan. stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm

40000	1054	0	1084	29	8670	-.2828929486612E+00	.2331868205685E-04
TOTAL	1054	0	1084	29	<b>86.70 (seconds)</b>	proc=	.00%
CG with angle restart. Oct. 6, 2006							
8	CG-DL Algorithm:Pressure Distribution Problem Dai and Liao (t=1). stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	1274	0	1570	295	12075	-.2828929227255E+00	.2803963705326E-04
TOTAL	1274	0	1570	295	120.75 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
9	CG-DL+ Algorithm:Pressure Distribution Problem Dai and Liao plus (t=1). stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	1417	0	1756	338	13528	-.2828929205948E+00	.3753177128139E-04
TOTAL	1417	0	1756	338	135.28 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
10	CG-SDC Algorithm:Pressure Distribution Problem Andrei - Sufficient Descent Condition. stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	1271	0	1406	134	11120	-.2828929420157E+00	.2397640342216E-04
TOTAL	1271	0	1406	134	111.20 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
11	CG-hDY Algorithm:Pressure Distribution Problem Hybrid Dai and Yuan. max(c*DY,min(HS,DY)). stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	1358	0	1748	389	13281	-.2828928568691E+00	.4362687608907E-04
TOTAL	1358	0	1748	389	132.81 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
12	CG-hDYz Algorithm:Pressure Distribution Problem Hybrid Dai and Yuan. max(0,min(HS,DY)). stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	1341	0	1736	394	13172	-.2828928419213E+00	.3215959955920E-04
TOTAL	1341	0	1736	394	131.72 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
13	CG-GN Algorithm:Pressure Distribution Problem Gilbert and Nocedal. stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	1149	0	1642	492	12076	-.2828927548479E+00	.3402344281648E-04
TOTAL	1149	0	1642	492	120.76 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
14	CG-HuS Algorithm:Pressure Distribution Problem Hu and Storey. max(0,min(PRP,FR)). stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	1547	0	2186	638	16205	-.2828928462808E+00	.3235100091232E-04
TOTAL	1547	0	2186	638	162.05 (seconds)	proc=	.00%



CG with angle restart. Oct. 6, 2006							
15	CG-TaS Algorithm:Pressure Distribution Problem Touat-Ahmed and Storey. (PRP <= FR). stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	1280	0	1803	522	13379	-.2828928189297E+00	.2503822999534E-04
TOTAL	1280	0	1803	522	133.79 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
16	CG-LS&CD Algorithm:Pressure Distribution Problem Hybrid Liu&Storey-Conjugate Descent. max(0,min(LS,CD)) stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	1421	0	1967	545	14637	-.2828928549869E+00	.2384040564539E-04
TOTAL	1421	0	1967	545	146.37 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
17	CG-BM Algorithm:Pressure Distribution Problem Birgin and Martinez (scaled Perry). Theta spectral. stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	1252	0	1581	328	12206	-.2828928828225E+00	.2785913192577E-04
TOTAL	1252	0	1581	328	122.06 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
18	CG-BM+ Algorithm:Pressure Distribution Problem Birgin and Martinez plus (scaled Perry). Theta spectral. stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	1035	0	1306	270	9964	-.2828929330066E+00	.1954683134793E-04
TOTAL	1035	0	1306	270	99.64 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
19	CG-sPRP Algorithm:Pressure Distribution Problem Scaled Polak-Ribiere and Polyak. Theta spectral. stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	918	0	1298	379	9561	-.2828928275694E+00	.3699997061549E-04
TOTAL	918	0	1298	379	95.61 (seconds)	proc=	.00%
CG with angle restart. Oct. 6, 2006							
20	CG-sFR Algorithm:Pressure Distribution Problem Scaled Fletcher and Reeves. Theta spectral. stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	2001	0	2090	89	16852	-.2828929080668E+00	.1588099274077E-03
TOTAL	2001	0	2090	89	168.52 (seconds)	proc=	.00%

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  $nx = 10$ ,  $ny = 10$ .



**Fig. A2.** Soluția problemei “Distribution in a Journal Bearing“ .  $nx = 10$ ,  $ny = 10$  .

### **Aplicația A3. “Optimal Design with Composite Materials Problem“**

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 and Moré [1991].

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 [1986] 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

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

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:

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

Problema considerată în Averick, Carter și Moré [1991] este aceea de minimizare a lui  $f_\lambda$  pentru o valoare fixată a lui  $\lambda$ . 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:

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

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  $1 \leq j \leq n_y + 1$ .

Considerînd  $\mu_1 = 1$  și  $\mu_2 = 2$ , în Tabelul A3 se prezintă performanța lui CGALL  $n_x = 200$  și  $n_y = 200$ .

**Tabelul A3.** Performanța lui CGALL. stoptest=1. angle restart.  
nx = 200, ny = 200.

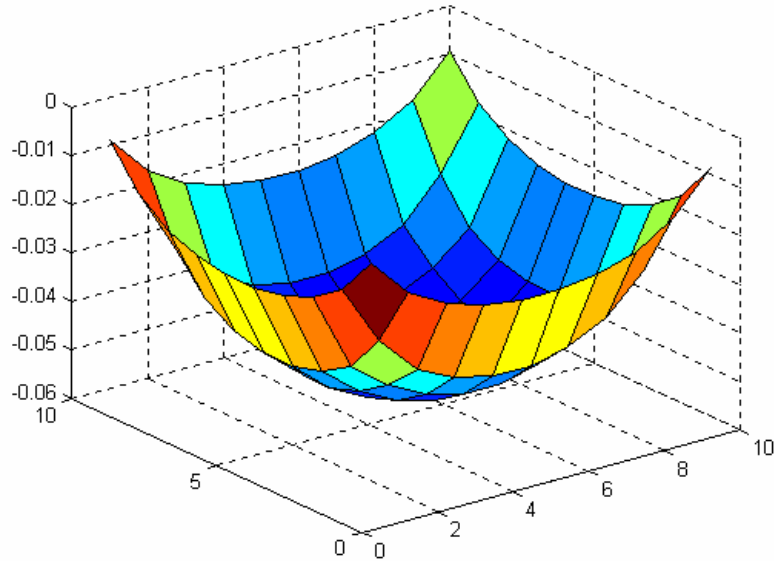
CG with angle restart. Oct. 7, 2006							
1 CG-HS Algorithm:Optimal Design with Composite Materials Function Hestenes and Stiefel. stoptest= 1							
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	1716	0	2010	292	19884	-.1138121978094E-01	.1688657672642E-04
TOTAL	1716	0	2010	292	198.84 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
2 CG-FR Algorithm:Optimal Design with Composite Materials Function Fletcher and Reeves. stoptest= 1							
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	2001	0	2097	96	21206	-.1050565370325E-01	.2687250149201E-01
TOTAL	2001	0	2097	96	212.06 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
3 CG-PRP Algorithm:Optimal Design with Composite Materials Function Polak-Ribiere and Polyak. stoptest= 1							

n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	2001	0	2681	680	25784	-.1136648443447E-01	.2185206771161E-03
TOTAL	2001	0	2681	680	257.84 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
4 CG-PRP+ Algorithm:Optimal Design with Composite Materials Function Polak-Ribiere and Polyak plus. stoptest= 1							
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	2001	0	2688	687	25832	-.1137386204106E-01	.3429778159997E-03
TOTAL	2001	0	2688	687	258.32 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
5 CG-CD Algorithm:Optimal Design with Composite Materials Function Conjugate Descent - Fletcher. stoptest= 1							
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	1648	3	1794	145	17895	-.1138128175981E-01	.2410200808003E-04
TOTAL	1648	3	1794	145	178.95 (seconds)	proc=	.18%
CG with angle restart. Oct. 7, 2006							
6 CG-LS Algorithm:Optimal Design with Composite Materials Function Liu and Storey. stoptest= 1							
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	2001	0	2582	581	25100	-.1137497408585E-01	.2073885942729E-03
TOTAL	2001	0	2582	581	251.00 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
7 CG-DY Algorithm:Optimal Design with Composite Materials Function Dai and Yuan. stoptest= 1							
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	2001	0	2013	12	20599	-.1124142173931E-01	.1806862697856E-01
TOTAL	2001	0	2013	12	205.99 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
8 CG-DL Algorithm:Optimal Design with Composite Materials Function Dai and Liao (t=1). stoptest= 1							
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	1470	0	1647	176	16436	-.1138111572805E-01	.3323356519196E-04
TOTAL	1470	0	1647	176	164.36 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
9 CG-DL+ Algorithm:Optimal Design with Composite Materials Function Dai and Liao plus (t=1). stoptest= 1							
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	1648	0	1844	195	18436	-.1138123805283E-01	.2108234399880E-04
TOTAL	1648	0	1844	195	184.36 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
10 CG-SDC Algorithm:Optimal Design with Composite Materials Function Andrei - Sufficient Descent Condition. stoptest= 1							
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	1374	0	1420	45	14436	-.1138128588727E-01	.3515202969329E-04

TOTAL	1374	0	1420	45	<b>144.36 (seconds)</b>	proc=	.00%
CG with angle restart. Oct. 7, 2006							
11 CG-hDY Algorithm:Optimal Design with Composite Materials Function Hybrid Dai and Yuan. max(c*DY,min(HS,DY)). stoptest= 1							
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	1878	0	2215	336	21849	-.1138126961334E-01	.2016410098584E-04
TOTAL	1878	0	2215	336	218.49 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
12 CG-hDYz Algorithm:Optimal Design with Composite Materials Function Hybrid Dai and Yuan. max(0,min(HS,DY)). stoptest= 1							
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	1878	0	2215	336	21845	-.1138126961334E-01	.2016410098584E-04
TOTAL	1878	0	2215	336	218.45 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
13 CG-GN Algorithm:Optimal Design with Composite Materials Function Gilbert and Nocedal. stoptest= 1							
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	2001	0	2630	629	25384	-.1137389673748E-01	.2902218382000E-03
TOTAL	2001	0	2630	629	253.84 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
14 CG-HuS Algorithm:Optimal Design with Composite Materials Function Hu and Storey. max(0,min(PRP,FR)). stoptest= 1							
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	2001	0	2641	640	25466	-.1137140083627E-01	.1793123082512E-03
TOTAL	2001	0	2641	640	254.66 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
15 CG-TaS Algorithm:Optimal Design with Composite Materials Function Touat-Ahmed and Storey. (PRP <= FR). stoptest= 1							
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	2001	0	2647	646	25546	-.1137507867734E-01	.3045918553877E-03
TOTAL	2001	0	2647	646	255.46 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
16 CG-LS&CD Algorithm:Optimal Design with Composite Materials Function Hybrid Liu&Storey-Conjugate Descent. max(0,min(LS,CD)) stoptest= 1							
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	2001	0	2560	559	24944	-.1137636331454E-01	.1144770497186E-03
TOTAL	2001	0	2560	559	249.44 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
17 CG-BM Algorithm:Optimal Design with Composite Materials Function Birgin and Martinez (scaled Perry). Theta spectral. stoptest= 1							
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	1756	0	2026	269	20165	-.1138117244604E-01	.2137827370116E-04
TOTAL	1756	0	2026	269	201.65 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							

18 CG-BM+ Algorithm:Optimal Design with Composite Materials Function Birgin and Martinez plus (scaled Perry). Theta spectral. stoptest= 1							
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	1577	0	1796	218	17929	-.1138120384015E-01	.3085689040924E-04
TOTAL	1577	0	1796	218	179.29 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
19 CG-sPRP Algorithm:Optimal Design with Composite Materials Function Scaled Polak-Ribiere and Polyak. Theta spectral. stoptest= 1							
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	2001	0	2679	678	25874	-.1134719165313E-01	.1597065466838E-02
TOTAL	2001	0	2679	678	258.74 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
20 CG-sFR Algorithm:Optimal Design with Composite Materials Function Scaled Fletcher and Reeves. Theta spectral. stoptest= 1							
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
40000	2001	0	2097	96	21336	-.1050565370333E-01	.2687250182713E-01
TOTAL	2001	0	2097	96	213.36 (seconds)	proc=	.00%

Figura A3 prezintă soluția problemei pentru discretizarea  $nx = 10, ny = 10$ .



**Fig. A3.** Soluția problemei “Optimal Design with Composite Materials“.  $nx = 10, ny = 10$ .

#### **Aplicația A4. “Inhomogeneous Superconductors. 1-dimensional Ginzburg-Landau Problem“**

Această problemă apare în rezolvarea ecuațiilor Ginzburg-Landau pentru superconductoare neomogene 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

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

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

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

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). 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

$$-d = t_1 < t_2 < \dots < t_n = d.$$

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

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

unde

$$f_i(v) = h_i \left\{ \frac{\alpha_i}{3} \frac{v_{i+1}^3 - v_i^3}{v_{i+1} - v_i} + \frac{\beta_i}{5} \frac{v_{i+1}^5 - v_i^5}{v_{i+1} - v_i} + \frac{\hbar}{4m} \left( \frac{v_{i+1} - v_i}{h_i} \right)^2 \right\},$$

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  $d = 3.2$  angstroms și temperatura  $T = 5$ , pentru 1000 de puncte de discretizare algoritmi de gradient conjugat implementați în CGALL furnizează rezultatele din Tabelul A4.

**Tabelul A4.** Performanța lui CGALL. stoptest=1. angle restart.  
 $n = 1000$ .

CG with angle restart. Oct. 7, 2006							
1	CG-HS Algorithm:Ginzburg-Landau (1-dimensional) Hestenes and Stiefel. stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
1000	14001	0	18155	4154	8814	-.2377192546376E-03	.3088007550039E-01
TOTAL	14001	0	18155	4154	88.14 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
2	CG-FR Algorithm:Ginzburg-Landau (1-dimensional) Fletcher and Reeves. stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
1000	14001	3	15145	1144	7792	-.5289434311113E-03	.8048443653448E-01
TOTAL	14001	3	15145	1144	77.92 (seconds)	proc=	.02%
CG with angle restart. Oct. 7, 2006							
3	CG-PRP Algorithm:Ginzburg-Landau (1-dimensional) Polak-Ribiere and Polyak. stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
1000	14001	1	20115	6114	9585	-.2241934486058E-03	.5657273403756E-02
TOTAL	14001	1	20115	6114	95.85 (seconds)	proc=	.01%
CG with angle restart. Oct. 7, 2006							
4	CG-PRP+ Algorithm:Ginzburg-Landau (1-dimensional) Polak-Ribiere and Polyak plus. stoptest= 1						Function

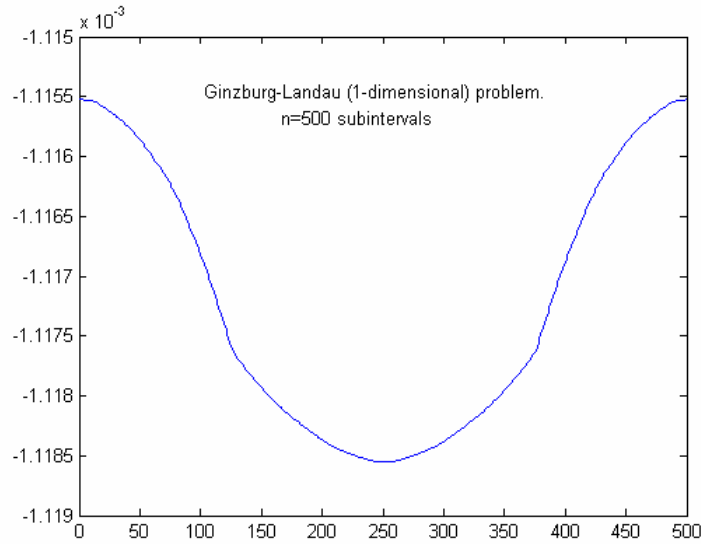
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
1000	14001	3	20082	6081	9775	-.2092403785916E-03	.4507718715303E-02
TOTAL	14001	3	20082	6081	97.75 (seconds)	proc=	.02%
CG with angle restart. Oct. 7, 2006							
5	CG-CD Algorithm:Ginzburg-Landau (1-dimensional) Conjugate Descent - Fletcher. stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
1000	14001	7	15022	1021	7696	-.2973343454095E-03	.2635576105963E-01
TOTAL	14001	7	15022	1021	76.96 (seconds)	proc=	.05%
CG with angle restart. Oct. 7, 2006							
6	CG-LS Algorithm:Ginzburg-Landau (1-dimensional) Liu and Storey. stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
1000	14001	0	19714	5713	9329	-.1868976052166E-03	.5214152188659E-02
TOTAL	14001	0	19714	5713	93.29 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
7	CG-DY Algorithm:Ginzburg-Landau (1-dimensional) Dai and Yuan. stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
1000	14001	7	14230	229	7270	-.2375450145690E+01	.1495732862068E+02
TOTAL	14001	7	14230	229	<b>72.70 (seconds)</b>	proc=	.05%
CG with angle restart. Oct. 7, 2006							
8	CG-DL Algorithm:Ginzburg-Landau (1-dimensional) Dai and Liao (t=1). stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
1000	14001	0	18291	4290	8896	-.1775548822355E-03	.1030458187818E-01
TOTAL	14001	0	18291	4290	88.96 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
9	CG-DL+ Algorithm:Ginzburg-Landau (1-dimensional) Dai and Liao plus (t=1). stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
1000	14001	0	18121	4120	8783	-.1794280430988E-03	.5591013192664E-02
TOTAL	14001	0	18121	4120	87.83 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
10	CG-SDC Algorithm:Ginzburg-Landau (1-dimensional) Andrei - Sufficient Descent Condition. stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
1000	14001	2	14333	332	7383	-.112223584022E-02	.1049927595854E+01
TOTAL	14001	2	14333	332	73.83 (seconds)	proc=	.01%
CG with angle restart. Oct. 7, 2006							
11	CG-hDY Algorithm:Ginzburg-Landau (1-dimensional) Hybrid Dai and Yuan. max(c*DY,min(HS,DY)). stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
1000	14001	0	18151	4150	8792	-.1794760029678E-03	.5420154809958E-02



TOTAL	14001	0	18151	4150	87.92 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
12	CG-hDYz Algorithm:Ginzburg-Landau (1-dimensional) Hybrid Dai and Yuan. max(0,min(HS,DY)). stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
1000	14001	0	18168	4167	8804	-.1789501521284E-03	.4441177761903E-02
TOTAL	14001	0	18168	4167	88.04 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
13	CG-GN Algorithm:Ginzburg-Landau (1-dimensional) Gilbert and Nocedal. stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
1000	14001	1	20006	6005	9433	-.3364423595679E-03	.4236531982274E-02
TOTAL	14001	1	20006	6005	94.33 (seconds)	proc=	.01%
CG with angle restart. Oct. 7, 2006							
14	CG-HuS Algorithm:Ginzburg-Landau (1-dimensional) Hu and Storey. max(0,min(PRP,FR)). stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
1000	14001	1	20002	6001	10099	-.2403447668737E-03	.1924754614831E-01
TOTAL	14001	1	20002	6001	100.99 (seconds)	proc=	.01%
CG with angle restart. Oct. 7, 2006							
15	CG-TaS Algorithm:Ginzburg-Landau (1-dimensional) Touat-Ahmed and Storey. (PRP <= FR). stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
1000	14001	0	19904	5903	9307	-.1860351521947E-03	.4093246180112E-02
TOTAL	14001	0	19904	5903	93.07 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
16	CG-LS&CD Algorithm:Ginzburg-Landau (1-dimensional) Hybrid Liu&Storey-Conjugate Descent. max(0,min(LS,CD)) stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
1000	14001	0	19514	5513	9354	-.1878380674015E-03	.4469163942246E-02
TOTAL	14001	0	19514	5513	93.54 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
17	CG-BM Algorithm:Ginzburg-Landau (1-dimensional) Birgin and Martinez (scaled Perry). Theta spectral. stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
1000	14001	0	17869	3868	8910	-.1761021393922E-03	.1080218678800E-01
TOTAL	14001	0	17869	3868	89.10 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
18	CG-BM+ Algorithm:Ginzburg-Landau (1-dimensional) Birgin and Martinez plus (scaled Perry). Theta spectral. stoptest= 1						Function
n	iter	irs	fgcnt	lsCnt	time(c)	fxnew	gnorm
1000	14001	0	17922	3921	8643	-.1824142630544E-03	.1403520027503E-01
TOTAL	14001	0	17922	3921	86.43 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							

19	CG-sPRP Algorithm:Ginzburg-Landau (1-dimensional) Scaled Polak-Ribiere and Polyak. Theta spectral. stoptest= 1						Function
n	iter	irs	fgcnt	lscent	time(c)	fxnew	gnorm
1000	14001	9	20093	6091	9544	-.2172388470274E-03	.8220352956666E-02
TOTAL	14001	9	20093	6091	95.44 (seconds)	proc=	.06%
CG with angle restart. Oct. 7, 2006							
20	CG-sFR Algorithm:Ginzburg-Landau (1-dimensional) Scaled Fletcher and Reeves. Theta spectral. stoptest= 1						Function
n	iter	irs	fgcnt	lscent	time(c)	fxnew	gnorm
1000	14001	2	14945	944	7563	-.3542339486115E-03	.3091610874304E+00
TOTAL	14001	2	14945	944	75.63 (seconds)	proc=	.01%

În Figura A4 se arată soluția problemei pentru  $n = 500$  puncte de discretizare.



**Fig. A4.** Soluția problemei “Ginzburg-Landau (1-dimensional)” .  $n = 500$  .

#### **Aplicația A5. “Steady State Combustion. Solid Fuel Ignition”**

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

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

unde  $f_\lambda : H_0^\perp(D) \rightarrow R$  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:

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

unde  $\Delta$  este operatorul Laplacian. Aris [1975], și Bebernes și Eberly [1989] discută această problemă în contextul combustiei solidelor.

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 pași  $h_x$  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

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

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

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

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

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  $1 \leq j \leq n_y + 1$ .

Tabelul A5 arată performanța lui CGALL pentru  $\lambda = 5$ ,  $n_x = 200$  și  $n_y = 200$ .

**Tabelul A5.** Performanța lui CGALL. stoptest=1. angle restart.  
nx = 200, ny = 200, lambda = 5.

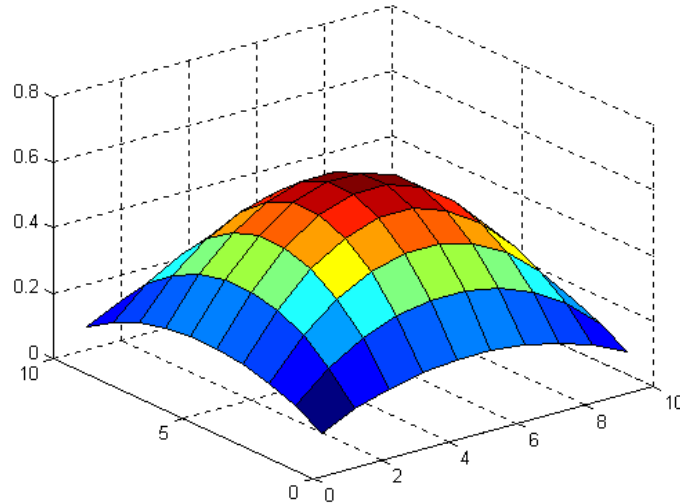
CG with angle restart. Oct. 7, 2006							
1	CG-HS Algorithm:Steady State Combustion Hestenes and Stiefel. stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	990	0	1282	291	15124	-.5611448088702E+01	.4241874165000E-04
TOTAL	990	0	1282	291	151.24 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
2	CG-FR Algorithm:Steady State Combustion Fletcher and Reeves. stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	2001	0	2125	124	25786	-.5611448508352E+01	.5026258157411E-04
TOTAL	2001	0	2125	124	257.86 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
3	CG-PRP Algorithm:Steady State Combustion Polak-Ribiere and Polyak. stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	841	0	1211	369	14009	-.5611447238752E+01	.6051631561179E-04
TOTAL	841	0	1211	369	140.09 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
4	CG-PRP+ Algorithm:Steady State Combustion Polak-Ribiere and Polyak plus. stoptest= 1						Function

n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	1280	0	1820	539	21115	-.5611446654176E+01	.5275409844863E-04
TOTAL	1280	0	1820	539	211.15 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
5 CG-CD Algorithm:Steady State Combustion Function Conjugate Descent - Fletcher. stoptest= 1							
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	1807	3	2022	214	24322	-.5611448511519E+01	.3739770994935E-04
TOTAL	1807	3	2022	214	243.22 (seconds)	proc=	.17%
CG with angle restart. Oct. 7, 2006							
6 CG-LS Algorithm:Steady State Combustion Function Liu and Storey. stoptest= 1							
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	937	0	1300	362	15119	-.5611447357498E+01	.4356711022336E-04
TOTAL	937	0	1300	362	151.19 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
7 CG-DY Algorithm:Steady State Combustion Function Dai and Yuan. stoptest= 1							
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	886	0	909	22	11137	-.5611448502855E+01	.4503303912575E-04
TOTAL	886	0	909	22	111.37 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
8 CG-DL Algorithm:Steady State Combustion Function Dai and Liao (t=1). stoptest= 1							
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	958	0	1089	130	13126	-.5611448275202E+01	.3638411180565E-04
TOTAL	958	0	1089	130	131.26 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
9 CG-DL+ Algorithm:Steady State Combustion Function Dai and Liao plus (t=1). stoptest= 1							
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	1001	0	1144	142	13789	-.5611448350552E+01	.4697740420196E-04
TOTAL	1001	0	1144	142	137.89 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
10 CG-SDC Algorithm:Steady State Combustion Function Andrei - Sufficient Descent Condition. stoptest= 1							
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	968	0	1028	59	12567	-.5611448460777E+01	.4236004138153E-04
TOTAL	968	0	1028	59	125.67 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
11 CG-hDY Algorithm:Steady State Combustion Function Hybrid Dai and Yuan. max(c*DY,min(HS,DY)). stoptest= 1							
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	865	0	1119	253	13174	-.5611448346802E+01	.4720759708192E-04

TOTAL	865	0	1119	253	131.74 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
12	CG-hDYz Algorithm:Steady State Combustion Hybrid Dai and Yuan. max(0,min(HS,DY)). stoptest= 1					Function	
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	982	0	1268	285	14933	-.5611448142923E+01	.3654265209350E-04
TOTAL	982	0	1268	285	149.33 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
13	CG-GN Algorithm:Steady State Combustion Gilbert and Nocedal. stoptest= 1					Function	
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	734	0	1034	299	11989	-.5611448184781E+01	.4206983899731E-04
TOTAL	734	0	1034	299	119.89 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
14	CG-HuS Algorithm:Steady State Combustion Hu and Storey. max(0,min(PRP,FR)). stoptest= 1					Function	
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	844	0	1200	355	13900	-.5611446855706E+01	.4701481684608E-04
TOTAL	844	0	1200	355	139.00 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
15	CG-TaS Algorithm:Steady State Combustion Touat-Ahmed and Storey. (PRP <= FR). stoptest= 1					Function	
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	774	0	1104	329	12796	-.5611447449967E+01	.4090301490055E-04
TOTAL	774	0	1104	329	127.96 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
16	CG-LS&CD Algorithm:Steady State Combustion Hybrid Liu&Storey-Conjugate Descent. max(0,min(LS,CD)) stoptest= 1					Function	
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	1080	0	1502	421	17477	-.5611448507849E+01	.4911707417145E-04
TOTAL	1080	0	1502	421	174.77 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
17	CG-BM Algorithm:Steady State Combustion Birgin and Martinez (scaled Perry). Theta spectral. stoptest= 1					Function	
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	892	0	1106	213	13156	-.5611447674984E+01	.4504763181220E-04
TOTAL	892	0	1106	213	131.56 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
18	CG-BM+ Algorithm:Steady State Combustion Birgin and Martinez plus (scaled Perry). Theta spectral. stoptest= 1					Function	
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	1206	0	1520	313	18055	-.5611448262696E+01	.4202960220530E-04
TOTAL	1206	0	1520	313	180.55 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							

19	CG-sPRP Algorithm:Steady State Combustion Scaled Polak-Ribiere and Polyak. Theta spectral. stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	1119	0	1596	476	18666	-.5611447735287E+01	.4039052157073E-04
TOTAL	1119	0	1596	476	186.66 (seconds)	proc=	.00%
CG with angle restart. Oct. 7, 2006							
20	CG-sFR Algorithm:Steady State Combustion Scaled Fletcher and Reeves. Theta spectral. stoptest= 1						Function
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
40000	1996	0	2106	109	25718	-.5611448507720E+01	.4766583196742E-04
TOTAL	1996	0	2106	109	257.18 (seconds)	proc=	.00%

Figura A5 arată soluția problemei combustiei solidelor (problema variațională Bratu) pentru  $n_x = 10$  și  $n_y = 10$ .



**Fig. A5.** Soluția problemei “Steady State Combustion. Solid Fuel Ignition“.  $n_x = 10$  și  $n_y = 10$ .

#### **Aplicația A6. “Lennard-Jones Cluster Problem”**

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

$$V_d(p) = \sum_{j=2}^n \sum_{i=1}^{j-1} v(\|p_j - p_i\|_2),$$

unde  $v: R \rightarrow R$  este funcția potențial între perechi de atomi. Funcția de potențial Lennard-Jones este definită ca:

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

Problema este de a determina o configurație (poziția celor  $n$  puncte) astfel încât funcția  $V_d$  să fie minimizată. Tabelul A6 arată performanța algoritmilor de gradient conjugat din CGALL pentru rezolvarea acestei probleme.

**Tabelul A6.** Performanța lui CGALL. stoptest =1. angle restart.  
ndim = 3, natoms =300.

CG with angle restart. Oct. 9, 2006							
1	CG-HS Algorithm:Jones Clusters (Molecular Conformation) Function Hestenes and Stiefel. stoptest= 1						
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
900	4001	19	118031	3630	137771	-.1828648172932E+04	.6300921292425E-05
TOTAL	4001	19	118031	3630	1377.71 (seconds)	proc=	.47%
CG with angle restart. Oct. 9, 2006							
2	CG-FR Algorithm:Jones Clusters (Molecular Conformation) Function Fletcher and Reeves. stoptest= 1						
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
900	4001	0	5343	1342	6431	-.1484475462042E+04	.2709037049905E+03
TOTAL	4001	0	5343	1342	64.31 (seconds)	proc=	.00%
CG with angle restart. Oct. 9, 2006							
3	CG-PRP Algorithm:Jones Clusters (Molecular Conformation) Function Polak-Ribiere and Polyak. stoptest= 1						
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
900	4001	5	121123	3773	141290	-.1821259280218E+04	.7796929698301E-05
TOTAL	4001	5	121123	3773	1412.90 (seconds)	proc=	.12%
CG with angle restart. Oct. 9, 2006							
4	CG-PRP+ Algorithm:Jones Clusters (Molecular Conformation) Function Polak-Ribiere and Polyak plus. stoptest= 1						
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
900	3758	2	114074	3532	132975	-.1827507111099E+04	.7198391467760E-05
TOTAL	3758	2	114074	3532	1329.75 (seconds)	proc=	.05%
CG with angle restart. Oct. 9, 2006							
5	CG-CD Algorithm:Jones Clusters (Molecular Conformation) Function Conjugate Descent - Fletcher. stoptest= 1						
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
900	4001	8	63726	2156	74484	-.1813486711772E+04	.2440479754219E-04
TOTAL	4001	8	63726	2156	744.84 (seconds)	proc=	.20%
CG with angle restart. Oct. 9, 2006							
6	CG-LS Algorithm:Jones Clusters (Molecular Conformation) Function Liu and Storey. stoptest= 1						
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
900	4001	1	121818	3764	142103	-.1822291051498E+04	.1130084178714E-04
TOTAL	4001	1	121818	3764	1421.03 (seconds)	proc=	.02%
CG with angle restart. Oct. 9, 2006							
7	CG-DY Algorithm:Jones Clusters (Molecular Conformation) Function Dai and Yuan. stoptest= 1						
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
900	4001	0	4014	13	4888	-.1487738758683E+04	.2170650073259E+03
TOTAL	4001	0	4014	13	48.88 (seconds)	proc=	.00%
CG with angle restart. Oct. 9, 2006							

---

8 CG-DL Algorithm:Jones Clusters (Molecular Conformation) Function  
Dai and Liao (t=1). stoptest= 1

n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
900	3948	19	118827	3662	138515	-.1823645142572E+04	.7829616839376E-05
<hr/>							
TOTAL	3948	19	118827	3662	1385.15 (seconds)	proc=	.48%

CG with angle restart. Oct. 9, 2006

9 CG-DL+ Algorithm:Jones Clusters (Molecular Conformation) Function  
Dai and Liao plus (t=1). stoptest= 1

n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
900	3556	0	107984	3294	125976	-.1819686885699E+04	.8190002586284E-05
<hr/>							
TOTAL	3556	0	107984	3294	1259.76 (seconds)	proc=	.00%

CG with angle restart. Oct. 9, 2006

10 CG-SDC Algorithm:Jones Clusters (Molecular Conformation) Function  
Andrei - Sufficient Descent Condition. stoptest= 1

n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
900	4001	5	27887	818	32621	-.1823961733981E+04	.9634668832112E-04
<hr/>							
TOTAL	4001	5	27887	818	326.21 (seconds)	proc=	.12%

CG with angle restart. Oct. 9, 2006

11 CG-hDY Algorithm:Jones Clusters (Molecular Conformation) Function  
Hybrid Dai and Yuan. max(c\*DY,min(HS,DY)). stoptest= 1

n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
900	4001	136	121226	3727	141310	-.1824671666606E+04	.3486765870985E-05
<hr/>							
TOTAL	4001	136	121226	3727	1413.10 (seconds)	proc=	3.40%

CG with angle restart. Oct. 9, 2006

12 CG-hDYz Algorithm:Jones Clusters (Molecular Conformation) Function  
Hybrid Dai and Yuan. max(0,min(HS,DY)). stoptest= 1

n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
900	3364	0	100965	3097	117796	-.1824671666606E+04	.6432170993111E-05
<hr/>							
TOTAL	3364	0	100965	3097	1177.96 (seconds)	proc=	.00%

CG with angle restart. Oct. 9, 2006

13 CG-GN Algorithm:Jones Clusters (Molecular Conformation) Function  
Gilbert and Nocedal. stoptest= 1

n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
900	2892	1	87431	2709	102018	-.1823373765520E+04	.5433142393346E-05
<hr/>							
TOTAL	2892	1	87431	2709	1020.18 (seconds)	proc=	.03%

CG with angle restart. Oct. 9, 2006

14 CG-HuS Algorithm:Jones Clusters (Molecular Conformation) Function  
Hu and Storey. max(0,min(PRP,FR)). stoptest= 1

n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
900	3389	1	102336	3180	119292	-.1823373765520E+04	.9414553472752E-05
<hr/>							
TOTAL	3389	1	102336	3180	1192.92 (seconds)	proc=	.03%

CG with angle restart. Oct. 9, 2006

15 CG-TaS Algorithm:Jones Clusters (Molecular Conformation) Function  
Touat-Ahmed and Storey. (PRP <= FR). stoptest= 1

---



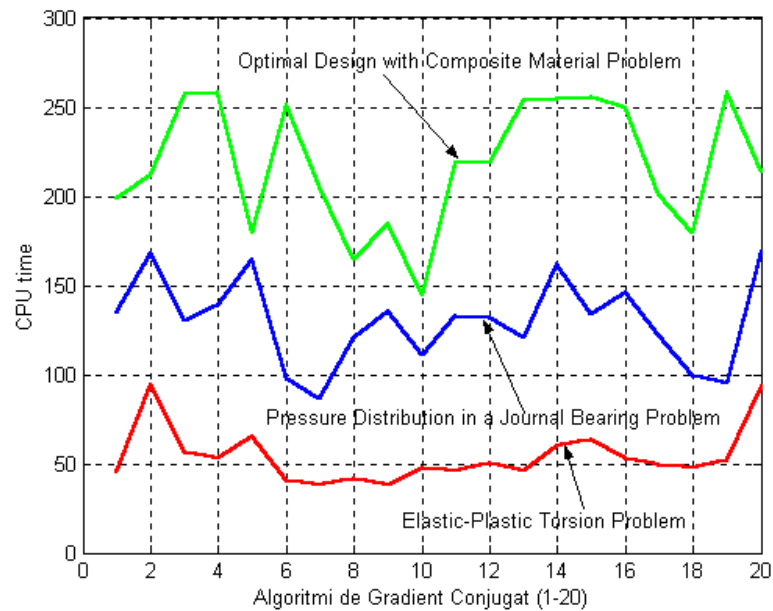
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
900	3553	1	107634	3344	125567	-.1823373765520E+04	.9281932376066E-05
TOTAL	3553	1	107634	3344	1255.67 (seconds)	proc=	.03%
CG with angle restart. Oct. 9, 2006							
16 CG-LS&CD Algorithm:Jones Clusters (Molecular Conformation) Function Hybrid Liu&Storey-Conjugate Descent. max(0,min(LS,CD)) stoptest= 1							
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
900	844	0	13351	525	15585	-.1824034524392E+04	.6343016811470E-05
TOTAL	844	0	13351	525	155.85 (seconds)	proc=	.00%
CG with angle restart. Oct. 9, 2006							
17 CG-BM Algorithm:Jones Clusters (Molecular Conformation) Function Birgin and Martinez (scaled Perry). Theta spectral. stoptest= 1							
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
900	1464	4	38626	1228	45036	-.1821045228841E+04	.5986429730137E-05
TOTAL	1464	4	38626	1228	450.36 (seconds)	proc=	.27%
CG with angle restart. Oct. 9, 2006							
18 CG-BM+ Algorithm:Jones Clusters (Molecular Conformation) Function Birgin and Martinez plus (scaled Perry). Theta spectral. stoptest= 1							
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
900	4001	1	61438	1837	71797	-.1821045228841E+04	.7935001534371E-05
TOTAL	4001	1	61438	1837	717.97 (seconds)	proc=	.02%
CG with angle restart. Oct. 9, 2006							
19 CG-sPRP Algorithm:Jones Clusters (Molecular Conformation) Function Scaled Polak-Ribiere and Polyak. Theta spectral. stoptest= 1							
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
900	3128	0	90373	2856	105457	-.1821259280218E+04	.7536453502620E-05
TOTAL	3128	0	90373	2856	1054.57 (seconds)	proc=	.00%
CG with angle restart. Oct. 9, 2006							
20 CG-sFR Algorithm:Jones Clusters (Molecular Conformation) Function Scaled Fletcher and Reeves. Theta spectral. stoptest= 1							
n	iter	irs	fgcnt	lscnt	time(c)	fxnew	gnorm
900	4001	0	5343	1342	6434	-.1484475376484E+04	.2709038450499E+03
TOTAL	4001	0	5343	1342	64.34 (seconds)	proc=	.00%

## Concluzii

Câteva concluzii sunt necesare.

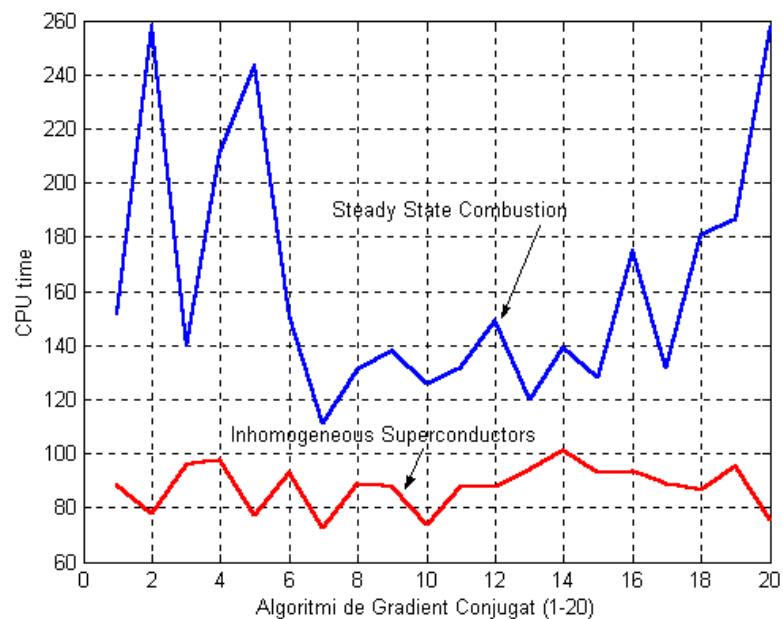
- 1) Metoda de gradient conjugat implementată în cei 20 de algoritmi prezentați în acest proiect se dovedește a fi capabilă să rezolve a multitudine de probleme de optimizare fără restricții.
- 2) Experimentele numerice intensive efectuate cu o colecție de 750 de probleme de optimizare fără restricții arată că algoritmul de gradient conjugat cu descendență suficientă, propus în [Andrei, 2006c] este “top performer” în această clasă.
- 3) Aplicațiile prezentată în această fază a proiectului, care sunt foarte complexe, arată că pentru anumite probleme algoritmii de gradient conjugat sunt competitivi în ceea ce privește rezolvarea

problemelor cu număr mare de variabile. Totuși, pentru alte probleme (vezi aplicația Lennard-Jones Cluster Problem) comportarea acestor algoritmi este mizerabilă. O analiză a rezultatelor prezentate în această secțiune arată că versiunea Dai-Yuan [1999] este câștigătoare. În imediata vecinătate se află versiunea noastră de gradient conjugat cu descendență suficientă. De exemplu, în Figura A6 se arată evoluția timpului cpu (secunde) în funcție de algoritmi de gradient conjugat considerați în acest studiu, corespunzătoare primelor trei aplicații.



**Fig. A6.** Evoluția timpului de calcul în funcție de algoritmi pentru primele 3 aplicații.

În Figura A7 se arată evoluția timpului CPU în funcție de algoritmi considerați pentru următoarele două aplicații.



**Fig. A7.** Evoluția timpului de calcul în funcție de algoritmi pentru următoarele 2 aplicații.

Ultima aplicație nu a mai fost ilustrată deoarece timpii de calcul sunt extrem de mari chiar pentru situația în care am considerat doar 300 de atomi. Aplicația a fost inclusă în acest studiu pentru a arăta impredictibilitatea comportării algoritmilor de gradient conjugat. Vedem că cei mai mici timp de calcul au fost obținuți cu varianta de gradient conjugat Dai-Yuan [1999].

4) Diferența esențială între versiunea noastră [Andrei, 2006c] și cea a lui Dai și Yuan [1999] este că în cazul algoritmului nostru direcția de deplasare este calculată sub forma  $d_{k+1} = -\theta_{k+1} g_{k+1} + \beta_k s_k$ , iar pentru versiunea Dai - Yuan aceasta se calculează simplu sub forma  $d_{k+1} = -g_{k+1} + \beta_k d_k$ . Prezența parametrului  $\theta_{k+1}$ , chiar dacă este calculat ca o aproximație scalară a inversei matricei Hessian, perturbă direcția de a fi una eficientă, mai ales în situația în care căutarea liniară se bazează pe condițiile Wolfe care, după cum știm, realizează o acuratețe de ordinul lui radical din epsilon mașină. De aceea efortul de cercetare trebuie dirijat către elaborarea de algoritmi care să utilizeze direcția  $d_{k+1} = -g_{k+1} + \beta_k d_k$  în care parametrul  $\beta_k$  este ales în așa mod încât cele două condiții esențiale: descendența suficientă și conjugarea să fie îndeplinite. În fond, algoritmi bazați pe direcția  $d_{k+1} = -g_{k+1} + \beta_k d_k$  sunt perturbări ale algoritmului pasului descendent.

5) În esența lor acești algoritmi **nu sunt bine cunoscuți**. Comportarea lor numerică este imprevizibilă. Slăbiciunea lor constă în faptul că aceștia nu includ nici un fel de informație de ordinul doi asupra funcției de minimizat. În comparație cu algoritmi quasi-Newton (cu memorie limitată) sau Newton trunchiat deseori acești algoritmi sunt inferiori. Totuși, trebuie să remarcăm aici simplitatea acestor algoritmi și mai ales ușurința în ceea ce privește implementarea lor în programe de calcul.

## Bibliografie

- Andrei, N., (1996) *Preliminary computational experience with ASLO an advanced system for linear optimization*. Technical Report No. AMOL-96-6, Research Institute for Informatics, Bucharest, August 22, 1996.
- Andrei, N., (1999a) *Programarea Matematică Avansată. Teorie, Metode Computaționale, Aplicații*. Editura Tehnică - București, 1999.
- Andrei, N., (1999b) *Programarea matematică. Metode de punct interior*. Editura Tehnică - București, 1999.
- Andrei, N., (2000a) *Optimizare fără restricții. Metode de direcții conjugate*. Editura MATRIXROM, București, 2000.
- Andrei, N., (2000b) *Metode de punct interior în optimizarea convexă*. Editura MATRIXROM, București, 2000.
- Andrei, N., (2001f) *Pachete de programe, modele și probleme de test pentru programarea matematică*. Editura MATRIXROM, București, 2001.
- Andrei, N., (2002) *Sisteme și pachete de programe pentru programarea matematică*. Editura Tehnică, București, 2002.
- Andrei, N., (2003) *Modele, Probleme de Test și Aplicații de Programare Matematică*, Editura Tehnică, București, 2003.
- Andrei, N., (2004a) *A new gradient descent method for unconstrained optimization*, ICI Technical Report, March 2004.
- Andrei, N., (2004b) *Scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization*. ICI Technical Report, September 2005.
- Andrei, N., (2005) Programul NUCLEU PN 05 13 01 05: *Critica convergenței și complexității algoritmilor de optimizare convexă fără restricții*. Faza 1: Formularea și verificarea unei teorii a convergenței și complexității optimizării convexe fără restricții - metode de gradient descendent, Martie 2005. Faza 2: Formularea și verificarea unei teorii a convergenței și complexității optimizării convexe fără restricții - metode de gradient conjugat, Decembrie 2005.
- Andrei, N., (2006) *An acceleration of gradient descent algorithm with backtracking for unconstrained optimization*. Numerical Algorithms, Volume 42, Number 1, pp. 63-73, May 2006.
- Andrei, N., (2006a) *Scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization*. Optimization Methods and Software, acceptată OMS 182164, 2006.

- Andrei, N., (2006b) *Scaled BFGS preconditioned conjugate gradient algorithm for unconstrained optimization*. Applied Mathematics Letters, acceptatã AML 2275, 2006.
- Andrei, N., (2006c) *A Conjugate Gradient Algorithm with Sufficient Descent Condition for Unconstrained Optimization*. ICI Technical Report, March 13, 2006.
- Andrei, N., (2006d) *Performance of conjugate gradient algorithms on some MINPACK-2 unconstrained optimization applications*. Studies in Informatics and Control, vol.15, No.2, June 2006, pp.145-168.
- Andrei, N., (2006e) *An acceleration of gradient descent algorithm with backtracking for unconstrained optimization*. Studies in Informatics and Control, vol.14, June 2005, pp.117-128.
- Aris, R., (1975) *The mathematical theory of diffusion and reaction in permeable catalysts*. Oxford, 1975.
- Averick, B.M., Carter, R.G. and Moré, J.J., (1991) *The MINPACK-2 test problem collection (preliminary version)*. Argonne National Laboratory, 9700 South Cass Avenue, Argonne, Illinois 60439, Technical Memorandum No.150, May 1991.
- Averick, B.M., R.G. Carter, J.J. Moré and G.L. Xue, (1992) “*The MINPACK-2 test problem collection*”, Argonne National Laboratory, Preprint MCS-P153-0692, June 1992.
- Bebernes, J., and Eberly, D., (1989) *Mathematical problems from combustion theory*. Applied Mathematical Sciences 83, Springer-Verlag, 1989.
- Bongartz, I., A.R. Conn, N.I.M. Gould and P.L. Toint, (1995) “*CUTE: constrained and unconstrained testing environments*”, ACM Trans. Math. Software, 21, pp.123-160, 1995.
- Cimatti, G. and 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.
- Dai, Y.H. and Yuan, Y., (1999) *A nonlinear conjugate gradient method with a strong global convergence property*. SIAM J. Optim., 10, pp.177-182, 1999.
- Elliot, C.M. and Ockendon, J.R., (1982) *Weak and variational methods for moving boundary problems*. Research Notes in Mathematics, vol.59, Pitman, 1982.
- Garner, J. and Benedek, R. (1990) *Solution of Ginzburg-Landau equations for inhomogeneous superconductors by nonlinear optimization*. Phys. Rev., B. 42, 1990, pp.376-385.
- Glowinski, R., (1984) *Numerical Methods for Nonlinear Variational Problems*. Springer-Verlag, 1984.
- Goodman, J., Kohn, R. and Reyna, L., (1986) *Numerical study of a relaxed variational problem from optimal design*. Comput. Methods Appl. Mech. Engrg., 57, 1986, pp.107-127.
- Hoare, M.R., (1979) *Structure and dynamics of simple microclusters*. Advances in Chemical Physics, 40, 1979, pp.49-135.
- Lin, Y. and Cryer, C.W., (1985) *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.
- Moré, J.J. and Toraldo, G., (1991) *On the solution of large quadratic programming problems with bound constraints*. SIAM J. Optimization, 1, 1991, pp.93-113.
- Northby, J.A., (1987) *Structure and binding of Lennard-Jones clusters:  $13 < n < 147$* . Journal of Chemical Physics, 87, 1987, pp.6166-6177.
- O’Leary, D.P. and Yang, W.H., (1978) *Elastoplastic torsion by quadratic programming*. Comput. Methods Appl. Mech. Engrg., 16, 1978, pp.361-368.

**Octombrie 16, 2006**