A spectral conjugate gradient method for unconstrained optimization *

Ernesto G. Birgin † José Mario Martínez ‡

August 20, 1999

Abstract

A family of scaled conjugate-gradient algorithms for large-scale unconstrained minimization is defined. The Perry, the Polak-Ribière and the Fletcher-Reeves formulae are compared using a spectral scaling derived from Raydan's spectral gradient optimization method. The best combination of formula, scaling and initial choice of step-length is compared against well known algorithms using a classical set of problems. An additional comparison involving an ill-conditioned estimation problem in Optics is presented.

Keywords. Unconstrained minimization, spectral gradient method, conjugate gradients.

AMS: 49M07, 49M10, 90C06, 65K.

*Abbreviated: Spectral conjugate gradient method

†Department of Computer Science, IME-USP, University of São Paulo, Rua do Matão, 1010 - Cidade Universitária, 05508-900, São Paulo SP, Brazil. This author was supported by FAPESP (Grants 98/07704-1 and 99/08029-9) and PRONEX-Optimization 76.79.1008-00. e-mail: egbirgin@ime.usp.br

‡Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil. This author was supported by FAPESP (Grant 90/3724-6), CNPq, FAEP-UNICAMP and PRONEX-Optimization 76.79.1008-00. e-mail: martinez@ime.unicamp.br

1 Introduction

In a recent paper [10] Raydan introduced the spectral gradient method (SGM) for potentially large-scale unconstrained optimization. The main feature of this method is that only gradient directions are used at each line search whereas a non-monotone strategy guarantees global convergence. Surprisingly, this algorithm outperforms sophisticated conjugate gradient algorithms in many problems. The numerical results in [1, 6, 7, 10] and others suggested us that spectral gradient and conjugate gradient ideas could be combined in order to obtain even more efficient algorithms.

Assume that $f : \mathbb{R}^n \to \mathbb{R}$ has continuous partial derivatives. The problem considered in this paper is

$$\text{Minimize } f(x), \quad x \in \mathbb{R}^n.$$

Algorithms for solving this problem are iterative. Here, the iterates will be denoted x_k, $k = 0, 1, 2, \ldots$ For each iteration we compute a search direction $d_k \in \mathbb{R}^n$ and successive iterates are obtained by means of

$$x_{k+1} = x_k + \alpha_k d_k.$$

Moreover, the directions are generated by

$$d_{k+1} = -\theta_k g_{k+1} + \beta_k s_k \tag{1}$$

for $k = 0, 1, 2, \ldots$, where g_k denotes $\nabla f(x_k)$, $x_0 \in \mathbb{R}^n$ is arbitrary and

$$d_0 = -\theta_0 g_0.$$

Assuming that x_k and x_{k+1} are two consecutive approximations, we denote:

$$s_k = x_{k+1} - x_k = \alpha_k d_k \quad \text{and} \quad y_k = g_{k+1} - g_k.$$

Suppose, for a moment, that f is quadratic and $H \equiv \nabla^2 f(x)$ is positive definite. This implies that $y_k \neq 0$. Therefore, the true minimizer x_* satisfies

$$x_* = x_{k+1} + d_*,$$

where

$$Hd_* = -g_{k+1}.$$

Pre-multiplying by s_k^T, this gives

$$s_k^T H d_* = -s_k^T g_{k+1}.$$

Therefore,

$$y_k^T d_* = -s_k^T g_{k+1}.$$

Thus, the hyper-plane

$$\mathcal{H}_k \equiv \{d \in I\!R^n \ \mid \ y_k^T d = -s_k^T g_{k+1}\}$$

contains the optimum increment d_*, which gives $x_* = x_{k+1} + d_*$. Observe that the null direction $d = 0$ belongs to $\mathcal{H}$ only if $s_k^T g_{k+1} = 0$ which is not our assumption at all.

By the discussion above, it is natural to impose, for the search direction d_{k+1},

$$d_{k+1} \in \mathcal{H}_k. \tag{2}$$

Then, by (1),

$$\beta_k = \frac{(\theta_k y_k - s_k)^T g_{k+1}}{s_k^T y_k}. \tag{3}$$

For $\theta_k = 1$ this formula was introduced by Perry in [9]. If we assume that $s_j^T g_{j+1} = 0$, $j = 0, 1, \ldots, k$, we obtain

$$\beta_k = \frac{\theta_k y_k^T g_{k+1}}{\alpha_k \theta_{k-1} g_k^T g_k}. \tag{4}$$

If $\theta_k = \theta_{k-1} = 1$ this is the classical Polak-Ribière formula. Finally, assuming that the successive gradients are orthogonal, we obtain the generalization of Fletcher-Reeves formula:

$$\beta_k = \frac{\theta_k g_{k+1}^T g_{k+1}}{\alpha_k \theta_{k-1} g_k^T g_k}. \tag{5}$$

In this paper, motivated by the success of the spectral gradient method, we decided to compare the classical choice $\theta_k = 1$ with the spectral gradient choice:

$$\theta_k = s_k^T s_k / s_k^T y_k. \tag{6}$$

In fact, the directions $d_k = -\theta_k g_k$ are the ones used by Raydan in his spectral gradient method. The parameter θ_k given by (6) is the inverse of the Rayleigh quotient

$$s_k^T [\int_0^1 \nabla^2 f(x_k + ts_k) dt] s_k / s_k^T s_k$$

which, of course, lies between the largest and the smallest eigenvalue of the Hessian average $\int_0^1 \nabla^2 f(x_k + ts_k)dt$.

After some numerical experimentation, we observed that the initial trial choice for the step-length α_k is a very important parameter that affects the algorithmic behavior. So, we decided to test two different alternatives for this choice.

This paper is organized as follows. In Section 2 we present the model algorithm, giving all the essential features of its implementation. In Section 3 we use the set of test problems of [10] to answer the following questions:

1. Is the choice (6) better than $\theta_k \equiv 1$?
2. Which is the best choice for β_k, among (3), (4) and (5)?
3. Which is the best initial choice for the step-length?

In Section 4 we compare the new algorithm against CONMIN (a popular conjugate-gradient code based on [11, 12]) and the spectral gradient method (SGM), using the same test functions of Section 3. In Section 5 we compare the new method against the spectral gradient algorithm using a real-life estimation problem in Optics. Conclusions are given in Section 6.

2 The algorithm

Keeping in mind the definitions of g_k, s_k and y_k given in the Introduction, we define the Scaled Conjugate Gradient method as follows.

Algorithm SCG

Assume that $x_0 \in I\!R^n$, $0 < \sigma < \gamma < 1$. Define $d_0 = -g_0$ and set $k \leftarrow 0$.

Step 1: If $g_k = 0$, terminate the execution of the algorithm.

Step 2: Compute (trying first $\alpha = \bar{\alpha}(k, d_k, d_{k-1}, \alpha_{k-1})$) $\alpha > 0$ such that

$$f(x_k + \alpha d_k) \leq f(x_k) + \sigma \alpha g_k^T d_k \tag{7}$$

and

$$\nabla f(x_k + \alpha d_k)^T d_k \geq \gamma g_k^T d_k. \tag{8}$$

Define $\alpha_k = \alpha$ and

$$x_{k+1} = x_k + \alpha_k d_k.$$

Step 3: Compute θ_k by (6) (or $\theta_k = 1$) and β_k by (3), (4) or (5).

Define

$$d = -\theta_k g_{k+1} + \beta_k s_k. \tag{9}$$

If

$$d^T g_{k+1} \leq -10^{-3} \|d\|_2 \|g_{k+1}\|_2 \tag{10}$$

define $d_{k+1} = d$. Otherwise, define

$$d_{k+1} = -\theta_k g_{k+1}.$$

Step 4: Set $k \leftarrow k + 1$ and go to Step 1.

It is well known (see [4, 5]) that a step-length α satisfying (7, 8) exists if f is bounded below along the direction d_k. We assume that we have an algorithm that either computes α with those conditions or detects that f is unbounded below. In this case, we say that SCG breaks at iteration k. In practice, we adopted the one-dimensional line search used in CONMIN (see [12]) for computing α.

The search direction d computed by (9) can fail to be a descent direction. This fact motivated several modifications of Perry's formula in [11]. In our algorithm, when the angle between d and $-g_{k+1}$ is not acute enough we "restart" the algorithm with the spectral gradient direction $-\theta_k g_{k+1}$. More sophisticated reasons for restarting have been proposed in the literature, but we are interested in the performance of an algorithm that uses this naive criterion, associated to the spectral gradient choice for restarts. Of course, the coefficient θ_k is always well defined and positive, since (8) implies that $s_k^T y_k > 0$.

Conditions (7), (8) and (10) are sufficient to prove global convergence of the algorithm under reasonable assumptions. If the gradient of f is Lipschitz-continuous and f is bounded below it can be proved that

$$\lim_{k\to\infty} \|g_k\| = 0.$$

See, for example, Theorem 3.1 of [8] and references therein. This implies that every limit point of a sequence generated by the algorithm is stationary.

3 Discussion of alternatives

In this section we use the test problems considered in [10] to answer the questions formulated in the Introduction. With this purpose, we consider the algorithm SCG with $\sigma = 10^{-4}$ and $\gamma = 0.5$.

For each choice of β_k (Perry (3), Polak-Ribière (4) or Fletcher-Reeves (5)) we have four methods:

M1: θ_k is computed by (6) and the initial choice of α is

$$\bar{\alpha}(k, d_k, d_{k-1}, \alpha_{k-1}) = \begin{cases} 1, & \text{if } k = 0 \\ \alpha_{k-1}\|d_{k-1}\|_2/\|d_k\|_2, & \text{otherwise;} \end{cases} \tag{11}$$

M2: θ_k is computed by (6) and $\bar{\alpha}(k, d_k, d_{k-1}, \alpha_{k-1}) \equiv 1$;

M3: $\theta_k \equiv 1$ and the initial α is computed as in (11);

M4: $\theta_k \equiv 1$ and $\bar{\alpha}(k, d_k, d_{k-1}, \alpha_{k-1}) \equiv 1$.

Tables 1, 2 and 3 display the performance of the algorithms described above. For each algorithm we state the number of function-gradient evaluations (FGE) and the functional value achieved at the approximate solution found ($f(x)$). For terminating the executions, we used, as in [10], the criterion

$$\|\nabla f(x_k)\|_2 \leq 10^{-6} \max\{1, |f(x_k)|\}.$$

The symbol NaN that appears in some executions of M2 and M4, means that the code tried to evaluate the function (or its gradient) at some point where it is not well defined. This can be avoided using some step-length control, but we decided not to do that in this comparative study.

Let f_i be the optimal functional value found by method M_i and f_j the optimal functional value found by M_j. We say that, in a particular problem, the performance of M_i was better than the performance of M_j if $f_i \leq f_j - 10^{-3}$ or if $|f_i - f_j| < 10^{-3}$ and the number of function-gradient evaluations of M_i was less than the number of function-gradient evaluation of M_j. The CPU time is not relevant for this comparison because all the alternatives are implemented in a unique code and the linear algebra per iteration is, basically, the same for all the methods.

The experiments were run in a SPARCstation Sun Ultra 1, with an UltraSPARC 64 bits processor, 167-MHz clock and 128-MBytes of RAM memory. All the codes considered in this paper were written in double precision Fortran and were compiled with the f77 compiler (SC 1.0 Fortran v1.4) using the optimization option -O4.

In Table 4 we find a summary of the comparison between the alternatives M1, M2, M3 and M4 of Perry, Polak-Ribière and Fletcher-Reeves. For example, the first entrance of this table should be read as follows: when comparing the performance of Perry-M1 and Perry-M2, Perry-M1 was better than Perry-M2 in 25 problems, worse in 14 problems and they had the same performance in 1 problem.

Problem		M1		M2		M3		M4	
		FGE	$f(x)$	FGE	$f(x)$	FGE	$f(x)$	FGE	$f(x)$
1	100	11	0.0000D+00	8	1.4211D−14	11	0.0000D+00	8	0.0000D+00
	1000	11	1.1369D−13	8	1.1369D−13	11	1.1369D−13	8	2.2737D−13
	10000	11	-1.8190D−12	11	-1.8190D−12	11	5.4570D−12	53	-3.6380D−12
2	100	63	5.0500D+02	79	5.0500D+02	63	5.0500D+02	83	5.0500D+02
	500	85	1.2525D+04	124	1.2525D+04	95	1.2525D+04	140	1.2525D+04
	1000	96	5.0050D+04	140	5.0050D+04	90	5.0050D+04	186	5.0050D+04
3	100	11	8.7540D−22	7	1.4666D−24	11	8.7540D−22	7	7.8758D−25
	1000	9	5.2302D−20	6	2.6191D−22	9	5.2302D−20	6	5.4108D−20
	10000	43	0.0000D+00	18	0.0000D+00	43	0.0000D+00	13	0.0000D+00
4	100	94	1.8410D−06	119	1.8410D−06	98	1.8410D−06	83	1.8410D−06
	1000	84	2.3338D−07	121	2.1479D−07	86	2.4019D−07	79	2.4705D−07
	10000	85	2.2553D−08	114	2.2105D−08	96	2.2560D−08	83	2.2370D−08
5	100	55	3.0248D−15	41	4.7262D−15	56	9.0436D−16	99	7.7355D−15
	1000	108	1.4078D+00	142	7.1253D−01	70	4.8690D−15	190	3.9707D−01
	3000	98	3.9707D−01	54	8.0081D−15	113	3.9707D−01	203	3.9707D−01
6	100	59	1.2885D−10	94	3.4514D−10	120	1.7172D−10	385	2.7329D−10
	1000	175	3.6787D−10	415	3.4794D−11	306	6.3862D−10	1794	7.1612D−10
	10000	823	3.7529D−10	2385	2.7418D−10	765	3.1394D−10	7879	1.4202D−10
7	100	54	7.1131D−24	107	1.9376D−25	49	1.7498D−20	118	3.8934D−15
	1000	60	7.8057D−23	127	4.3936D−26	50	1.1695D−16	113	1.9968D−27
	10000	61	3.2663D−21	93	3.6097D−21	53	2.1468D−17	134	6.4256D−23
8	100	152	9.0249D−04	158	9.0249D−04	110	9.0249D−04	255	9.0249D−04
	1000	104	9.6862D−03	132	9.6862D−03	81	9.6862D−03	579	9.6862D−03
	10000	96	9.9002D−02	89	9.9002D−02	98	9.9002D−02	650	9.9002D−02
9	100	190	2.8146D−15	242	3.5976D−15	181	2.8217D−15	163	1.0748D−15
	1000	746	1.5807D−15	711	8.5520D−16	700	1.5831D−15	615	4.3545D−16
10	100	29	1.0563D−19	22	4.6474D−21	43	5.1365D−21	310	3.6212D−19
	1000	82	1.5639D−18	216	9.3215D−23	114	9.5874D−23	56	NaN
11	100	174	1.4167D−09	359	1.9077D−10	149	1.1705D−10	246	4.2275D−10
	1000	163	1.0096D−10	554	2.9010D−09	98	5.7400D−09	672	3.4859D−09
12	100	694	1.0000D+00	744	1.0000D+00	519	1.0000D+00	3836	1.0000D+00
	500	1913	1.0000D+00	3340	1.0000D+00	2109	1.0000D+00	16888	1.0000D+00
13	100	32	1.0909D+02	27	1.0909D+02	39	1.0909D+02	45	1.0909D+02
	1000	31	1.1082D+03	22	1.1082D+03	34	1.1082D+03	48	1.1082D+03
	10000	23	1.1099D+04	19	1.1099D+04	36	1.1099D+04	49	1.1099D+04
14	100	85	1.1965D+04	129	1.1965D+04	65	1.1965D+04	527	1.1965D+04
	1000	43	1.2147D+05	77	1.2147D+05	112	1.2147D+05	257	1.2147D+05
	10000	41	1.2165D+06	118	1.2165D+06	38	1.2165D+06	293	1.2165D+06
15	100	120	3.2370D−16	79	3.7810D+02	87	3.7810D+02	386	7.8770D+00
	1000	104	5.3242D−15	166	3.9379D+00	77	3.9306D+03	364	3.9228D+03

Table 1: Performance of Perry.

		M1		M2		M3		M4	
Problem		FGE	$f(x)$	FGE	$f(x)$	FGE	$f(x)$	FGE	$f(x)$
1	100	13	-1.4211D−14	11	5.6843D−14	13	0.0000D+00	11	0.0000D+00
	1000	13	-1.1369D−13	11	3.4106D−13	13	-2.2737D−13	11	0.0000D+00
	10000	13	0.0000D+00	12	0.0000D+00	13	0.0000D+00	11	0.0000D+00
2	100	68	5.0500D+02	81	5.0500D+02	68	5.0500D+02	91	5.0500D+02
	500	111	1.2525D+04	123	1.2525D+04	116	1.2525D+04	163	1.2525D+04
	1000	124	5.0050D+04	140	5.0050D+04	128	5.0050D+04	187	5.0050D+04
3	100	11	8.7540D−22	8	4.7974D−19	11	8.7540D−22	5	NaN
	1000	9	5.2302D−20	8	3.1454D−20	9	5.2302D−20	5	NaN
	10000	43	0.0000D+00	74	NaN	43	0.0000D+00	5	NaN
4	100	117	2.4054D−06	121	1.8410D−06	117	2.4054D−06	95	1.8410D−06
	1000	103	2.3339D−07	108	2.2664D−07	103	2.3339D−07	111	2.1427D−07
	10000	116	2.2264D−08	121	2.1983D−08	104	2.2265D−08	94	2.2679D−08
5	100	51	9.3293D−15	52	1.0384D−14	51	9.3293D−15	116	5.4871D−15
	1000	122	7.1253D−01	112	7.1253D−01	124	7.1253D−01	235	7.1253D−01
	3000	116	3.9707D−01	61	5.4181D−15	119	3.9707D−01	195	3.9707D−01
6	100	75	8.1586D−11	110	3.3852D−11	75	8.1586D−11	361	1.8057D−10
	1000	270	4.9229D−10	519	4.6150D−10	268	5.1372D−10	1714	8.0629D−11
	10000	1117	1.1931D−11	3469	3.5720D−10	1064	1.8114D−10	8308	7.1085D−10
7	100	59	2.7563D−16	113	1.6326D−21	59	2.7563D−16	121	1.2111D−22
	1000	80	6.1238D−17	79	1.0247D−20	66	3.3072D−17	106	5.4745D−17
	10000	52	9.0002D−17	68	8.8774D−16	52	8.9995D−17	129	4.0085D−18
8	100	176	9.0249D−04	164	9.0249D−04	193	9.0249D−04	359	9.0249D−04
	1000	160	9.6862D−03	140	9.6862D−03	150	9.6862D−03	626	9.6862D−03
	10000	100	9.9002D−02	125	9.9002D−02	104	9.9002D−02	840	9.9002D−02
9	100	232	6.6821D−15	241	1.8589D−15	235	1.2005D−14	163	1.1622D−15
	1000	975	1.5198D−14	697	1.3306D−15	977	3.0449D−15	613	5.3448D−16
10	100	29	1.0511D−19	49	7.3915D−21	29	1.0511D−19	934	5.3293D−23
	1000	43	2.2841D−23	482	9.4991D−23	43	2.6460D−23	105	4.3118D+80
11	100	373	3.2342D−09	301	1.1965D−09	520	6.1000D−10	274	6.1994D−10
	1000	364	4.6163D−10	270	3.6148D−11	322	9.5744D−11	266	1.4964D−11
12	100	868	1.0000D+00	1030	1.0000D+00	777	1.0000D+00	4825	1.0000D+00
	500	3034	1.0000D+00	3990	1.0000D+00	3092	1.0000D+00	25400	1.0000D+00
13	100	39	1.0909D+02	24	1.0909D+02	39	1.0909D+02	34	1.0909D+02
	1000	36	1.1082D+03	22	1.1082D+03	36	1.1082D+03	47	1.1082D+03
	10000	30	1.1099D+04	28	1.1099D+04	30	1.1099D+04	32	1.1099D+04
14	100	75	1.1965D+04	129	1.1965D+04	77	1.1965D+04	263	1.1965D+04
	1000	62	1.2147D+05	100	1.2147D+05	72	1.2147D+05	430	1.2147D+05
	10000	65	1.2165D+06	43	1.2165D+06	65	1.2165D+06	225	1.2165D+06
15	100	66	3.8597D+02	80	3.7810D+02	63	3.8597D+02	343	3.9379D+00
	1000	77	3.9267D+03	76	3.9267D+03	77	3.9267D+03	364	3.9228D+03

Table 2: Performance of Polak-Ribière.

Problem		M1 FGE	M1 $f(x)$	M2 FGE	M2 $f(x)$	M3 FGE	M3 $f(x)$	M4 FGE	M4 $f(x)$
1	100	13	2.7001D−13	12	9.9476D−14	13	2.5580D−13	13	2.8422D−13
	1000	15	-1.1369D−13	13	1.1369D−13	15	-2.2737D−13	14	3.4106D−13
	10000	15	1.8190D−12	13	0.0000D+00	15	1.8190D−12	118	-1.8190D−12
2	100	87	5.0500D+02	131	5.0500D+02	87	5.0500D+02	104	5.0500D+02
	500	135	1.2525D+04	203	1.2525D+04	135	1.2525D+04	169	1.2525D+04
	1000	150	5.0050D+04	235	5.0050D+04	150	5.0050D+04	326	5.0050D+04
3	100	11	8.7540D−22	9	1.6427D−19	11	8.7540D−22	5	NaN
	1000	9	5.2302D−20	8	4.8953D−20	9	5.2302D−20	5	NaN
	10000	43	0.0000D+00	27	NaN	43	0.0000D+00	5	NaN
4	100	9108	2.0486D−06	514	1.8410D−06	9171	2.0439D−06	706	1.8410D−06
	1000	489	2.3349D−07	478	2.2725D−07	477	2.3349D−07	1066	2.2725D−07
	10000	298	2.1434D−08	583	1.4611D−08	214	2.1433D−08	946	2.1368D−08
5	100	94	6.1146D−15	88	6.5015D−15	94	6.1146D−15	112	7.3569D−15
	1000	580	3.9707D−01	489	3.9707D−01	357	3.9707D−01	2496	3.9707D−01
	3000	373	1.2611D−14	332	1.4078D−14	368	1.3015D−14	259	3.9707D−01
6	100	72	7.3343D−11	87	6.7439D−11	72	7.3343D−11	335	2.8331D−10
	1000	181	1.5513D−10	653	5.5624D−11	181	1.5532D−10	1764	1.6422D−10
	10000	749	9.3124D−11	4604	8.0323D−11	754	6.4120D−11	12168	4.3333D−11
7	100	243	6.3738D−13	327	2.7771D−13	202	1.5148D−17	3258	4.5981D−16
	1000	154	6.2276D−15	336	1.6205D−14	149	8.8002D−15	1825	3.0003D−14
	10000	208	6.7539D−14	1356	3.6998D−14	184	1.2493D−14	1339	3.4468D−13
8	100	1624	9.0249D−04	271	9.0249D−04	1636	9.0249D−04	1480	9.0249D−04
	1000	751	9.6862D−03	221	9.6862D−03	751	9.6862D−03	582	9.6862D−03
	10000	489	9.9002D−02	729	9.9002D−02	478	9.9002D−02	1913	9.9002D−02
9	100	454	2.1190D−15	255	1.3748D−15	454	2.1190D−15	163	1.1146D−15
	1000	2205	3.8403D−16	740	4.8037D−16	2205	3.8402D−16	613	5.3410D−16
10	100	29	1.0512D−19	55	5.3176D−19	29	1.0511D−19	929	1.9734D−24
	1000	43	1.6625D−23	503	1.4175D−22	43	1.3398D−23	56	NaN
11	100	232	2.1380D−09	704	4.9650D−10	913	4.6145D−10	16189	1.0111D−06
	1000	934	2.4712D−09	519	2.6710D−09	487	8.9763D−10	440	1.8797D−09
12	100	9975	1.0060D+02	21614	1.0000D+00	9975	1.0060D+02	8439	1.0000D+00
	500	9804	2.9171D+02	68651	1.8331D+02	9804	2.9171D+02	135876	2.1733D+02
13	100	50	1.0909D+02	42	1.0909D+02	50	1.0909D+02	50	1.0909D+02
	1000	40	1.1082D+03	52	1.1082D+03	40	1.1082D+03	28	1.1082D+03
	10000	33	1.1099D+04	19	1.1099D+04	33	1.1099D+04	64	1.1099D+04
14	100	63	1.1965D+04	3650	1.1965D+04	65	1.1965D+04	2565	1.1965D+04
	1000	22	1.2147D+05	1790	1.2147D+05	22	1.2147D+05	531	1.2147D+05
	10000	31	1.2165D+06	44	1.2165D+06	37	1.2165D+06	457	1.2165D+06
15	100	9041	3.9379D+00	794	3.7810D+02	8860	3.9379D+00	377	7.8770D+00
	1000	358	7.8770D+00	525	3.9379D+00	111	7.8770D+00	2217	7.8770D+00

Table 3: Performance of Fletcher-Reeves.

	Perry	Polak-Ribière	Fletcher-Reeves
M1 vs. M2	25-14-01	21-19-00	20-20-00
M1 vs. M3	19-14-07	12-08-20	06-11-23
M1 vs. M4	29-11-00	29-11-00	28-10-02
M2 vs. M3	16-23-01	18-22-00	18-22-00
M2 vs. M4	26-10-04	29-09-02	28-11-01
M3 vs. M4	27-13-00	28-12-00	28-10-02

Table 4: Comparison among M1, M2, M3 and M4.

Comparing the best alternatives of each conjugate-gradient formula, we conclude that: Perry-M1 beat Polak-Ribière-M1: 31-05-04, Perry-M1 beat Fletcher-Reeves-M3: 29-07-04, and Polak-Ribière-M1 beat Fletcher-Reeves-M3: 24-10-06.

4 Comparisons with CONMIN and SGM

The experiments in Section 3 seem to indicate that the best scaled conjugate-gradient formula is Perry's (3) with the spectral choice (6) of θ_k and the initial choice (11) of the step-length. Accordingly, we compared this method against CONMIN [11] and the spectral gradient method. We used the original (Fortran) codes of SGM and CONMIN. SGM was used with the parameters recommended by Raydan [10]. This algorithm uses $3n + O(1)$ real storage positions whereas CONMIN and SCG require $5n + O(1)$ real positions.

The results are given in Table 5. We report function evaluations (FE), gradient evaluations (GE), function-gradient evaluations (FGE), best function value ($f(x)$) and CPU time (Time). Since the methods compared here do not have the same linear algebra overhead, it makes sense to compare computer times. Considering CPU-time, we observe that Perry-M1 beats both CONMIN and SGM (31-03-06 and 23-12-05, respectively).

		SGM				CONMIN			Perry-M1		
Problem		FE	GE	Time	$f(x)$	FGE	Time	$f(x)$	FGE	Time	$f(x)$
1	100	8	8	0.00	0.0000D+00	38	0.01	1.6058D−11	11	0.00	0.0000D+00
	1000	8	8	0.01	-1.1369D−13	38	0.05	1.5154D−10	11	0.02	1.1369D−13
	10000	8	8	0.11	1.8190D−12	38	0.51	1.4734D−09	11	0.17	-1.8190D−12
2	100	57	52	0.01	5.0500D+02	81	0.02	5.0500D+02	63	0.01	5.0500D+02
	500	80	74	0.07	1.2525D+04	127	0.16	1.2525D+04	85	0.08	1.2525D+04
	1000	91	82	0.14	5.0050D+04	145	0.37	5.0050D+04	96	0.18	5.0050D+04
3	100	3	3	0.00	1.8795D−23	7	0.00	1.4066D−07	11	0.00	8.7540D−22
	1000	4	4	0.00	1.3346D−23	38	0.04	8.1381D−18	9	0.01	5.2302D−20
	10000	56	53	1.82	0.0000D+00	38	1.79	4.6837D−20	43	1.29	0.0000D+00
4	100	80	76	0.04	2.4054D−06	108	0.07	1.8410D−06	94	0.06	1.8410D−06
	1000	104	91	0.43	2.2558D−07	112	0.63	2.2664D−07	84	0.54	2.3338D−07
	10000	99	89	3.64	2.1659D−08	126	6.57	2.2674D−08	85	5.14	2.2553D−08
5	100	34	34	0.01	1.1369D−14	67	0.01	3.0081D−14	55	0.00	3.0248D−15
	1000	40	40	0.04	4.3612D−15	169	0.29	3.9707D−01	108	0.09	1.4078D+00
	3000	45	44	0.13	2.0021D−14	71	0.38	1.8685D−14	98	0.25	3.9707D−01
6	100	111	106	0.02	5.5889D−10	99	0.02	6.7141D−10	59	0.00	1.2885D−10
	1000	364	296	0.26	1.4567D−09	320	0.47	1.3921D−10	175	0.16	3.6787D−10
	10000	1751	1351	11.54	1.0295D−09	937	16.14	5.3219D−11	823	8.67	3.7529D−10
7	100	91	69	0.01	3.4615D−17	47	0.01	2.9286D−12	54	0.00	7.1131D−24
	1000	118	93	0.06	1.4427D−20	73	0.06	1.4111D−15	60	0.03	7.8057D−23
	10000	92	70	0.46	1.9663D−17	69	0.64	1.4479D−14	61	0.34	3.2663D−21
8	100	49	48	0.01	9.0249D−04	65	0.01	9.0249D−04	152	0.01	9.0249D−04
	1000	57	57	0.05	9.6862D−03	55	0.06	9.6862D−03	104	0.06	9.6862D−03
	10000	70	70	0.56	9.9002D−02	3	0.02	1.1114D+23	96	0.58	9.9002D−02
9	100	191	167	0.03	2.4820D−16	161	0.03	4.9988D−15	190	0.01	2.8146D−15
	1000	1152	878	0.78	1.6416D−14	613	1.08	6.1288D−16	746	0.65	1.5807D−15
10	100	38	38	0.01	3.1061D−29	29	0.00	2.8874D−18	29	0.00	1.0563D−19
	1000	68	66	0.07	1.6362D−25	62	0.07	1.4308D−20	82	0.06	1.5639D−18
11	100	988	740	0.17	1.1325D−09	95	0.02	1.0019D−09	174	0.01	1.4167D−09
	1000	1851	1345	2.19	7.9284D−09	87	0.17	2.0417D−09	163	0.09	1.0096D−10
12	100	1886	1429	0.22	1.0000D+00	516	0.09	1.0000D+00	694	0.04	1.0000D+00
	500	5896	4452	2.03	1.0000D+00	2180	1.64	1.0000D+00	1913	0.63	1.0000D+00
13	100	26	26	0.01	1.0909D+02	27	0.01	1.0909D+02	32	0.00	1.0909D+02
	1000	23	23	0.02	1.1082D+03	23	0.03	1.1082D+03	31	0.02	1.1082D+03
	10000	21	21	0.19	1.1099D+04	19	0.27	1.1099D+04	23	0.16	1.1099D+04
14	100	587	438	0.09	1.1965D+04	27	0.01	1.1965D+04	85	0.01	1.1965D+04
	1000	391	288	0.38	1.2147D+05	25	0.04	1.2147D+05	43	0.04	1.2147D+05
	10000	154	119	1.50	1.2165D+06	23	0.41	1.2165D+06	41	0.39	1.2165D+06
15	100	84	81	0.02	3.8597D+02	53	0.01	3.7810D+02	120	0.01	3.2370D−16
	1000	87	80	0.09	7.8770D+00	69	0.11	3.9267D+03	104	0.06	5.3242D−15

Table 5: Performance of SGM, CONMIN and Perry-M1.

5 A parameter estimation problem in Optics

In a recent works, the spectral gradient method has been successfully used for a hard inverse problem that consists on the estimation of optical parameters of thin films using transmission data. See [1].

The data of the problem is a set of N transmission observations for different wavelengths $((\lambda_i, T_i^{obs}), i = 1, 2, \ldots, N)$ and the objective is to recover the true thickness and the refractive and absorption parameters of the film. The unconstrained formulation introduced in [1] is as follows:

$$\text{Minimize } \sum_{i=1}^{N} [T(\lambda_i, d, n_i, \alpha_i) - T_i^{obs}]^2.$$

The transmission T of a thin absorbing film on a transparent substrate depends on a complicate formula that involves the thickness d, the refractive index $n(\lambda)$, the absorption coefficient $\alpha(\lambda)$ and the wavelength λ. The detailed description of the problem, as well as the pointwise unconstrained optimization strategy of solution, can be found in [1]. See, also, [3].

In [2] it has been pointed out that the main reason for slow convergence of the spectral gradient method in critical problems is local ill-conditioning at the solution. This complicating characteristic appears very strongly in this problem, because large variations of absorption coefficients produce an almost null variation of the transmission in the transparent zone of the spectrum. Therefore, the problem is practically under-determined on that zone. On the other hand, the spectral gradient method is very efficient for finding reasonable suboptimal solutions. For this reason, we conjectured that the spectral conjugate gradient variation presented in this paper could combine rapid approach to a solution basin and fast local convergence.

In our experiments we considered the five films analyzed in [1]. The physically acceptable results of the estimation procedure were obtained in [1] using 30000 iterations of the spectral gradient method. Here we used, as stopping criterion for SCG (Perry-M1), the inequality $f(x^k) < f_{Raydan}$ where f_{Raydan} is the minimum value reached by SGM. In Table 6 we give the results. We report IT (number of iterations), FE (functional evaluations),

	SGM				SCG			
Problem	IT	FE	Time	f_{Raydan}	IT	FGE	Time	$f(x^k)$
1	30000	35825	45.0”	6.929605E−07	3605	6184	7.7”	6.926210E−07
2	30000	35568	45.8”	2.203053E−07	6798	11092	14.1”	2.201913E−07
3	30000	38113	47.9”	6.224862E−06	7344	13471	17.5”	6.224860E−06
4	30000	35687	44.6”	1.365270E−06	10356	17938	22.3”	1.365184E−06
5	30000	36290	46.3”	2.120976E−07	7611	13205	16.5”	2.066100E−07

Table 6: Optics problems.

FGE (function-gradient evaluations) and Time (CPU time). Observe that SCG arrives to the same solution of SGM using between one third and one half of the computer time used by the spectral gradient method.

6 Final remarks

In the classical paper [11], Perry's basic idea was modified in order to overcome the lack of positive definiteness of the matrix that, implicitly, defines the search direction. As a result, the algorithmic framework of CONMIN was obtained. In this paper we followed a different direction, motivated by the necessity of preserving the nice geometrical properties of Perry's direction. On one hand, we observed that scaling the gradient by means of the spectral parameter of [10] is worthwhile and, on the other hand, we detected that the initial choice of the step-length crucially affects the practical behavior of the method. With the proper parameters, Perry's algorithm clearly outperforms Polak-Ribière and Fletcher-Reeves and is competitive with CONMIN and Raydan's [10] method.

Moreover, as observed by Raydan [10], the spectral gradient method needs preconditioning in ill-conditioned problems in a more dramatic way that conjugate-gradient methods do. This is the reason why, in the hard inverse problem studied in Section 5, SGM is outperformed by the M1 version of Perry's method.

Acknowledgements

We are indebted to two anonymous referees whose comments helped to improve the quality of this paper.

References

[1] Birgin E. G., Chambouleyron I. and Martínez J. M. (1999) Estimation of optical constants of thin films using unconstrained optimization, *Journal of Computational Physics* 151: 862–880.

[2] Birgin E. G., Martínez J. M. and Raydan M. (2000) Nonmonotone spectral projected gradient methods on convex sets, *SIAM Journal on Optimization* 10: 1196–1211.

[3] Chambouleyron I., Martínez J. M., Moretti A. C. and Mulato M. (1997) The retrieval of the optical constants and the thickness of thin films from transmission spectra, *Applied Optics* 36: 8238–8247.

[4] Dennis Jr. J. E. and Schnabel R. B. (1983) *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs.

[5] Fletcher R. (1987) *Practical Methods of Optimization* (2nd edition), John Wiley and Sons, Chichester, New York, Brisbane, Toronto and Singapore.

[6] Glunt W., Hayden T. L. and Raydan M. (1993) Molecular conformations from distance matrices, *Journal of Computational Chemistry* 14: 114–120.

[7] Luengo F., Raydan M., Glunt W. and Hayden T. L. (1996) Preconditioned spectral gradient method for unconstrained optimization problems, Technical Report 96-08, Escuela de Computación, Facultad de

Ciencias, Universidad Central de Venezuela, 47002 Caracas 1041-A, Venezuela.

[8] Nocedal, J. (1992) Theory of algorithms for unconstrained optimization, *Acta Numerica 1992*: 199–242.

[9] Perry A. (1978) A modified conjugate gradient algorithm, *Operations Research* 26: 1073–1078.

[10] Raydan M. (1997) The Barzilain and Borwein gradient method for the large unconstrained minimization problem, *SIAM Journal on Optimization* 7: 26–33.

[11] Shanno D. F. (1978) Conjugate gradient methods with inexact searches, *Mathematics of Operations Research* 3: 244–256.

[12] Shanno D. F. and Phua K. H. (1976) Minimization of unconstrained multivariate functions, *ACM Transactions on Mathematical Software* 2: 87–94.