Inexact Spectral Projected Gradient Methods on Convex Sets

Ernesto G. Birgin *   José Mario Martínez †   Marcos Raydan ‡

March 26, 2003

Abstract

A new method is introduced for large scale convex constrained optimization. The general model algorithm involves, at each iteration, the approximate minimization of a convex quadratic on the feasible set of the original problem and global convergence is obtained by means of nonmonotone line searches. A specific algorithm, the Inexact Spectral Projected Gradient method (ISPG), is implemented using inexact projections computed by Dykstra's alternating projection method and generates interior iterates. The ISPG method is a generalization of the Spectral Projected Gradient method (SPG), but can be used when projections are difficult to compute. Numerical results for constrained least-squares rectangular matrix problems are presented.

Key words: Convex constrained optimization, projected gradient, nonmonotone line search, spectral gradient, Dykstra's algorithm.

AMS Subject Classification: 49M07, 49M10, 65K, 90C06, 90C20.

1 Introduction

We consider the problem

$$\text{Minimize } f(x) \quad \text{subject to} \quad x \in \Omega, \tag{1}$$

where Ω is a closed convex set in $I\!R^n$. Throughout this paper we assume that f is defined and has continuous partial derivatives on an open set that contains Ω.

The Spectral Projected Gradient (SPG) method [6, 7] was recently proposed for solving (1), especially for large-scale problems since the storage requirements are minimal. This

*Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, Rua do Matão 1010 Cidade Universitária, 05508-090 São Paulo, SP - Brazil (egbirgin@ime.usp.br). Sponsored by FAPESP (Grants 01/04597-4 and 02/00094-0), CNPq (Grant 300151/00-4) and Pronex.

†Departamento de Matemática Aplicada, IMECC-UNICAMP, CP 6065, 13081-970 Campinas SP, Brazil (martinez@ime.unicamp.br). Sponsored by FAPESP (Grant 01/04597-4), CNPq and FAEP-UNICAMP.

‡Departamento de Computación, Facultad de Ciencias, Universidad Central de Venezuela, Ap. 47002, Caracas 1041-A, Venezuela (mraydan@reacciun.ve). Sponsored by the Center of Scientific Computing at UCV.

method has proved to be effective for very large-scale convex programming problems. In [7] a family of location problems was described with a variable number of variables and constraints. The SPG method was able to solve problems of this family with up to 96254 variables and up to 578648 constraints in very few seconds of computer time. The computer code that implements SPG and produces the mentioned results is published [7] and available. More recently, in [5] an active-set method which uses SPG to leave the faces was introduced, and bound-constrained problems with up to 10^7 variables were solved.

The SPG method is related to the practical version of Bertsekas [3] of the classical gradient projected method of Goldstein, Levitin and Polyak [21, 25]. However, some critical differences make this method much more efficient than its gradient-projection predecessors. The main point is that the first trial step at each iteration is taken using the spectral steplength (also known as the Barzilai-Borwein choice) introduced in [2] and later analyzed in [9, 19, 27] among others. The spectral step is a Rayleigh quotient related with an average Hessian matrix. For a review containing the more recent advances on this special choice of steplength see [20]. The second improvement over traditional gradient projection methods is that a nonmonotone search must be used [10, 22]. This feature seems to be essential to preserve the nice and nonmonotone behaviour of the iterates produced by single spectral gradient steps.

The reported efficiency of the SPG method in very large problems motivated us to introduce the inexact-projection version of the method. In fact, the main drawback of the SPG method is that it requires the exact projection of an arbitrary point of $I\!R^n$ onto Ω at every iteration.

Projecting onto Ω is a difficult problem unless Ω is an *easy* set (i.e. it is easy to project onto it) as a box, an affine subspace, a ball, etc. However, for many important applications, Ω is not an easy set and the projection can only be achieved inexactly. For example, if Ω is the intersection of a finite collection of closed and convex easy sets, cycles of alternating projection methods could be used. This sequence of cycles could be stopped prematurely leading to an inexact iterative scheme. In this work we are mainly concerned with extending the machinery developed in [6, 7] for the more general case in which the projection onto Ω can only be achieved inexactly.

In Section 2 we define a general model algorithm and prove global convergence. In Section 3 we introduce the ISPG method and we describe the use of Dykstra's alternating projection method for obtaining inexact projections onto closed and convex sets. In Section 4 we present numerical experiments and in Section 5 we draw some conclusions.

2 A general model algorithm and its global convergence

We say that a point $x \in \Omega$ is *stationary*, for problem (1), if

$$g(x)^T d \geq 0 \qquad (2)$$

for all $d \in I\!R^n$ such that $x + d \in \Omega$.

In this work $\|\cdot\|$ denotes the 2-norm of vectors and matrices, although in some cases it can be replaced by an arbitrary norm. We also denote $g(x) = \nabla f(x)$ and $I\!N = \{0, 1, 2, \ldots\}$.

Let $\mathcal{B}$ be the set of $n \times n$ positive definite matrices such that $\|B\| \leq L$ and $\|B^{-1}\| \leq L$. Therefore, $\mathcal{B}$ is a compact set of $I\!R^{n\times n}$. In the spectral gradient approach, the matrices will be diagonal. However, the algorithm and theorem that we present below are quite general. The matrices B_k may be thought as defining a sequence of different metrics in $I\!R^n$ according to which we perform projections. For this reason, we give the name "Inexact Variable Metric" to the method introduced below.

Algorithm 2.1: Inexact Variable Metric Method

Assume $\eta \in (0,1]$, $\gamma \in (0,1)$, $0 < \sigma_1 < \sigma_2 < 1$, M a positive integer. Let $x_0 \in \Omega$ be an arbitrary initial point. We denote $g_k = g(x_k)$ for all $k \in I\!N$. Given $x_k \in \Omega$, $B_k \in \mathcal{B}$, the steps of the $k-$th iteration of the algorithm are:

Step 1. *Compute the search direction*

Consider the subproblem

$$\text{Minimize } Q_k(d) \quad \text{subject to} \quad x_k + d \in \Omega, \tag{3}$$

where

$$Q_k(d) = \frac{1}{2} d^T B_k d + g_k^T d.$$

Let $\bar{d}_k$ be the minimizer of (3). (This minimizer exists and is unique by the strict convexity of the subproblem (3), but we will see later that we do not need to compute it.)

Let d_k be such that $x_k + d_k \in \Omega$ and

$$Q_k(d_k) \leq \eta \, Q_k(\bar{d}_k). \tag{4}$$

If $d_k = 0$, stop the execution of the algorithm declaring that x_k is a stationary point.

Step 2. *Compute the steplength*

Set $\alpha \leftarrow 1$ and $f_{\max} = \max\{f(x_{k-j+1}) \mid 1 \leq j \leq \min\{k+1, M\}\}$.
If

$$f(x_k + \alpha d_k) \leq f_{\max} + \gamma \alpha g_k^T d_k, \tag{5}$$

set $\alpha_k = \alpha$, $x_{k+1} = x_k + \alpha_k d_k$ and finish the iteration. Otherwise, choose $\alpha_{\text{new}} \in [\sigma_1 \alpha, \sigma_2 \alpha]$, set $\alpha \leftarrow \alpha_{\text{new}}$ and repeat test (5).

Remark. In the definition of Algorithm 2.1 the possibility $\eta = 1$ corresponds to the case in which the subproblem (3) is solved exactly.

Lemma 2.1. *The algorithm is well defined.*

Proof. Since Q_k is strictly convex and the domain of (3) is convex, the problem (3) has a unique solution $\bar{d}_k$. If $\bar{d}_k = 0$ then $Q_k(\bar{d}_k) = 0$. Since d_k is a feasible point of (3), and, by (4), $Q_k(d_k) \leq 0$, it turns out that $d_k = \bar{d}_k$. Therefore, $d_k = 0$ and the algorithm stops.

If $\bar{d}_k \neq 0$, then, since $Q_k(0) = 0$ and the solution of (3) is unique, it follows that $Q_k(\bar{d}_k) < 0$. Then, by (4), $Q_k(d_k) < 0$. Since Q_k is convex and $Q_k(0) = 0$, it follows that d_k is a descent direction for Q_k, therefore, $g_k^T d_k < 0$. So, for $\alpha > 0$ small enough,

$$f(x_k + \alpha d_k) \leq f(x_k) + \gamma \alpha g_k^T d_k.$$

Therefore, the condition (5) must be satisfied if α is small enough. This completes the proof. □

Theorem 2.1. *Assume that the level set $\{x \in \Omega \mid f(x) \leq f(x_0)\}$ is bounded. Then, either the algorithm stops at some stationary point x_k, or every limit point of the generated sequence is stationary.*

The proof of Theorem 2.1 is based on the following lemmas.

Lemma 2.2. *Assume that the sequence generated by Algorithm 2.1 stops at x_k. Then, x_k is stationary.*

Proof. If the algorithm stops at some x_k, we have that $d_k = 0$. Therefore, $Q_k(d_k) = 0$. Then, by (4), $Q_k(\bar{d}_k) = 0$. So, $\bar{d}_k = 0$. Therefore, for all $d \in I\!R^n$ such that $x_k + d \in \Omega$ we have $g_k^T d \geq 0$. Thus, x_k is a stationary point. □

For the remaining results of this section we assume that the algorithm does not stop. So, infinitely many iterates $\{x_k\}_{k \in I\!N}$ are generated and, by (5), $f(x_k) \leq f(x_0)$ for all $k \in I\!N$. Thus, under the hypothesis of Theorem 2.1, the sequence $\{x_k\}_{k \in I\!N}$ is bounded.

Lemma 2.3. *Assume that $\{x_k\}_{k \in I\!N}$ is a sequence generated by Algorithm 2.1. Define, for all $j = 1, 2, 3, \ldots$,*

$$V_j = \max\{f(x_{jM-M+1}), f(x_{jM-M+2}) \ldots, f(x_{jM})\},$$

and $\nu(j) \in \{jM - M + 1, jM - M + 2, \ldots, jM\}$ such that

$$f(x_{\nu(j)}) = V_j.$$

Then,

$$V_{j+1} \leq V_j + \gamma \alpha_{\nu(j+1)-1}\, g_{\nu(j+1)-1}^T\, d_{\nu(j+1)-1}. \tag{6}$$

for all $j = 1, 2, 3, \ldots$.

Proof. We will prove by induction on ℓ that for all $\ell = 1, 2, \ldots, M$ and for all $j = 1, 2, 3, \ldots$,

$$f(x_{jM+\ell}) \leq V_j + \gamma \alpha_{jM+\ell-1} g_{jM+\ell-1}^T d_{jM+\ell-1} < V_j. \tag{7}$$

By (5) we have that, for all $j \in I\!N$,

$$f(x_{jM+1}) \leq V_j + \gamma\alpha_{jM}\ g_{jM}^T\ d_{jM} < V_j,$$

so (7) holds for $\ell = 1$.

Assume, as the inductive hypothesis, that

$$f(x_{jM+\ell'}) \leq V_j + \gamma\alpha_{jM+\ell'-1} g_{jM+\ell'-1}^T d_{jM+\ell'-1} < V_j \tag{8}$$

for $\ell' = 1, \ldots, \ell$.

Now, by (5), and the definition of V_j, we have that

$$f(x_{jM+\ell+1}) \leq \max_{1\leq t\leq M} \{f(x_{jM+\ell+1-t}\} + \gamma\alpha_{jM+\ell}\ g_{jM+\ell}^T\ d_{jM+\ell}$$

$$= \max\{f(x_{(j-1)M+\ell+1}), \ldots, f(x_{jM+\ell})\} + \gamma\alpha_{jM+\ell}\ g_{jM+\ell}^T\ d_{jM+\ell}$$

$$\leq \max\{V_j, f(x_{jM+1}), \ldots, f(x_{jM+\ell})\} + \gamma\alpha_{jM+\ell}\ g_{jM+\ell}^T\ d_{jM+\ell}.$$

But, by the inductive hypothesis,

$$\max\{f(x_{jM+1}), \ldots, f(x_{jM+\ell})\} < V_j,$$

so,

$$f(x_{jM+\ell+1}) \leq V_j + \gamma\alpha_{jM+\ell}\ g_{jM+\ell}^T\ d_{jM+\ell} < V_j.$$

Therefore, the inductive proof is complete and, so, (7) is proved. Since $\nu(j+1) = jM + \ell$ for some $\ell \in \{1, \ldots, M\}$, this implies the desired result. □

From now on, we define

$$K = \{\nu(1) - 1,\ \nu(2) - 1,\ \nu(3) - 1,\ \ldots\},$$

where $\{\nu(j)\}$ is the sequence of indices defined in Lemma 2.3. Clearly,

$$\nu(j) < \nu(j+1) \leq \nu(j) + 2M \tag{9}$$

for all $j = 1, 2, 3, \ldots$.

Lemma 2.4.

$$\lim_{k\in K} \alpha_k Q_k(\bar{d}_k) = 0.$$

Proof. By (6), since f is continuous and bounded below,

$$\lim_{k\in K} \alpha_k g_k^T d_k = 0. \tag{10}$$

But, by (4),

$$0 > Q_k(d_k) = \frac{1}{2} d_k^T B_k d_k + g_k^T d_k \geq g_k^T d_k\ \forall\ k \in I\!N.$$

So,

$$0 > \eta Q_k(\bar{d}_k) \geq Q_k(d_k) \geq g_k^T d_k \ \forall\ k \in I\!N.$$

Therefore,

$$0 > \eta \alpha_k Q_k(\bar{d}_k) \geq \alpha_k Q_k(d_k) \geq \alpha_k g_k^T d_k \ \forall\ k \in K.$$

Hence, by (10),

$$\lim_{k \in K} \alpha_k Q_k(\bar{d}_k) = 0,$$

as we wanted to prove. □

Lemma 2.5. *Assume that $K_1 \subset I\!N$ is a sequence of indices such that*

$$\lim_{k \in K_1} x_k = x_* \in \Omega$$

and

$$\lim_{k \in K_1} Q_k(\bar{d}_k) = 0.$$

Then, x_* is stationary.

Proof. By the compactness of $\mathcal{B}$ we can extract a subsequence of indices $K_2 \subset K_1$ such that

$$\lim_{k \in K_2} B_k = B,$$

where B also belongs to $\mathcal{B}$.

We define

$$Q(d) = \frac{1}{2} d^T B d + g(x_*)^T d \ \ \forall\ d \in I\!R^n.$$

Suppose that there exists $\hat{d} \in I\!R^n$ such that $x_* + \hat{d} \in \Omega$ and

$$Q(\hat{d}) < 0. \tag{11}$$

Define

$$\hat{d}_k = x_* + \hat{d} - x_k \ \ \forall\ \ k \in K_2.$$

Clearly, $x_k + \hat{d}_k \in \Omega$ for all $k \in K_2$. By continuity, since $\lim_{k \in K_2} x_k = x_*$, we have that

$$\lim_{k \in K_2} Q_k(\hat{d}_k) = Q(\hat{d}) < 0. \tag{12}$$

But, by the definition of $\bar{d}_k$, we have that $Q_k(\bar{d}_k) \leq Q_k(\hat{d}_k)$, therefore, by (12),

$$Q_k(\bar{d}_k) \leq \frac{Q(\hat{d})}{2} < 0$$

for $k \in K_2$ large enough. This contradicts the fact that $\lim_{k \in K_2} Q_k(\bar{d}_k) = 0$. The contradiction came from the assumption that $\hat{d}$ with the property (11) exists. Therefore, $Q(d) \geq 0$ for all $d \in I\!R^n$ such that $x_* + d \in \Omega$. Therefore, $g(x_*)^T d \geq 0$ for all $d \in I\!R^n$ such that $x_* + d \in \Omega$. So, x_* is stationary. □

Lemma 2.6. *$\{d_k\}_{k\in I\!N}$ is bounded.*

Proof. For all $k \in I\!N$,

$$\frac{1}{2}d_k^T B_k d_k + g_k^T d_k < 0,$$

therefore, by the definition of $\mathcal{B}$,

$$\frac{1}{2L}\|d_k\|^2 + g_k^T d_k < 0 \;\forall\; k \in I\!N.$$

So, by Cauchy-Schwarz inequality

$$\|d_k\|^2 < -2Lg_k^T d_k \le 2L\|g_k\| \, \|d_k\| \;\forall\; k \in I\!N.$$

Therefore,

$$\|d_k\| < 2L\|g_k\| \;\forall\; k \in I\!N.$$

Since $\{x_k\}_{k\in I\!N}$ is bounded and f has continuous derivatives, $\{g_k\}_{k\in I\!N}$ is bounded. Therefore, the set $\{d_k\}_{k\in I\!N}$ is bounded. □

Lemma 2.7. *Assume that $K_3 \subset I\!N$ is a sequence of indices such that*

$$\lim_{k\in K_3} x_k = x_* \in \Omega \;\; \text{and} \;\; \lim_{k\in K_3} \alpha_k = 0.$$

Then,

$$\lim_{k\in K_3} Q_k(\bar{d}_k) = 0 \tag{13}$$

and, hence, x_* is stationary.

Proof. Suppose that (13) is not true. Then, for some infinite set of indices $K_4 \subset K_3$, $Q_k(\bar{d}_k)$ is bounded away from zero.

Now, since $\alpha_k \to 0$, for $k \in K_4$ large enough there exists α_k' such that $\lim_{k\in K_4} \alpha_k' = 0$, and (5) does not hold when $\alpha = \alpha_k'$. So,

$$f(x_k + \alpha_k' d_k) > \max\{f(x_{k-j+1}) \mid 1 \le j \le \min\{k+1, M\}\} + \gamma\alpha_k' g_k^T d_k.$$

Hence,

$$f(x_k + \alpha_k' d_k) > f(x_k) + \gamma\alpha_k' g_k^T d_k$$

for all $k \in K_4$. Therefore,

$$\frac{f(x_k + \alpha_k' d_k) - f(x_k)}{\alpha_k'} > \gamma g_k^T d_k$$

for all $k \in K_4$. By the mean value theorem, there exists $\xi_k \in [0,1]$ such that

$$g(x_k + \xi_k\alpha_k' d_k)^T d_k > \gamma g_k^T d_k \tag{14}$$

for all $k \in K_4$. Since the set $\{d_k\}_{k\in K_4}$ is bounded, there exists a sequence of indices $K_5 \subset K_4$ such that $\lim_{k\in K_5} d_k = d$ and $B \in \mathcal{B}$ such that $\lim_{k\in K_5} B_k = B$. Taking limits for $k \in K_5$ in both sides of (14), we obtain $g(x_*)^T d \geq \gamma g(x_*)^T d$. This implies that $g(x_*)^T d \geq 0$. So,

$$\frac{1}{2} d^T B d + g(x_*)^T d \geq 0.$$

Therefore,

$$\lim_{k\in K_5} d_k^T B_k d_k + g_k^T d_k = 0.$$

By (4) this implies that $\lim_{k\in K_5} Q_k(\bar{d}_k) = 0$. This contradicts the assumption that $Q_k(\bar{d}_k)$ is bounded away from zero for $k \in K_4$. Therefore, (13) is true. Thus the hypothesis of Lemma 2.5 holds, with K_3 replacing K_1. So, by Lemma 2.5, x_* is stationary. □

Lemma 2.8. *Every limit point of* $\{x_k\}_{k\in K}$ *is stationary.*

Proof. By Lemma 2.4, the thesis follows applying Lemma 2.5 and Lemma 2.7. □

Lemma 2.9. *Assume that* $\{x_k\}_{k\in K_6}$ *converges to a stationary point* x_*. *Then,*

$$\lim_{k\in K_6} Q_k(\bar{d}_k) = \lim_{k\in K_6} Q_k(d_k) = 0. \tag{15}$$

Proof. Assume that $Q_k(d_k)$ does not tend to 0 for $k \in K_6$. Then, there exists $\varepsilon > 0$ and an infinite set of indices $K_7 \subset K_6$ such that

$$Q_k(d_k) = \frac{1}{2} d_k^T B_k d_k + g_k^T d_k \leq -\varepsilon < 0.$$

Since $\{d_k\}_{k\in \mathbb{N}}$ is bounded and $x_k + d_k \in \Omega$, extracting an appropriate subsequence we obtain $d \in \mathbb{R}^n$ and $B \in \mathcal{B}$ such that $x_* + d \in \Omega$ and

$$\frac{1}{2} d^T B d + g(x_*)^T d \leq -\varepsilon < 0.$$

Therefore, $g(x_*)^T d < 0$, which contradicts the fact that x_* is stationary. Then,

$$\lim_{k\in K_6} Q_k(d_k) = 0.$$

So, by (4), the thesis is proved. □

Lemma 2.10. *Assume that* $\{x_k\}_{k\in K_8}$ *converges to some stationary point* x_*. *Then,*

$$\lim_{k\in K_8} \|d_k\| = \lim_{k\in K_8} \|x_{k+1} - x_k\| = 0.$$

Proof. Suppose that $\lim_{k\in K_8} \|d_k\| = 0$ is not true. By Lemma 2.6, $\{d_k\}_{k\in K_8}$ is bounded. So, we can take a subsequence $K_9 \subset K_8$ and $\varepsilon > 0$ such that

$$x_k + d_k \in \Omega \ \ \forall\, k \in K_9,$$

$$\|d_k\| \geq \varepsilon > 0 \;\; \forall\, k \in K_9,$$

$$\lim_{k \in K_9} B_k = B \in \mathcal{B}, \;\; \lim_{k \in K_9} x_k = x_* \in \Omega \tag{16}$$

and

$$\lim_{k \in K_9} d_k = d \neq 0. \tag{17}$$

By (15), (16), (17), we have that

$$\frac{1}{2} d^T B d + g(x_*)^T d = 0.$$

So, $g(x_*)^T d < 0$. Since x_* is stationary, this is impossible. □

Lemma 2.11. *For all* $r = 0, 1, \ldots, 2M$,

$$\lim_{k \in K} Q_k(\bar{d}_{k+r}) = 0. \tag{18}$$

Proof. By Lemma 2.10, the limit points of $\{x_k\}_{k \in K}$ are the same as the limit points of $\{x_{k+1}\}_{k \in K}$. Then, by Lemma 2.9,

$$\lim_{k \in K} Q_k(\bar{d}_{k+1}) = 0.$$

and, by Lemma 2.10,

$$\lim_{k \in K} \|d_{k+1}\| = 0.$$

So, by an inductive argument, we get

$$\lim_{k \in K} Q_k(\bar{d}_{k+r}) = 0$$

for all $r = 0, 1, \ldots, 2M$. □

Lemma 2.12.

$$\lim_{k \in I\!N} Q_k(\bar{d}_k) = 0. \tag{19}$$

Proof. Suppose that (19) is not true. Then there exists a subsequence $K_{10} \subset I\!N$ such that $Q_k(\bar{d}_k)$ is bounded away from zero for $k \in K_{10}$. But, by (9), every $k \in I\!N$ can be written as

$$k = k' + r$$

for some $k' \in K$ and $r \in \{0, 1, \ldots, 2M\}$. In particular, this happens for all $k \in K_{10}$. Since $\{0, 1, \ldots, 2M\}$ is finite, there exists a subsequence $K_{11} \subset K_{10}$ such that for all $k \in K_{11}$, $k = k' + r$ for some $k' \in K$ and *the same* $r \in \{0, 1, \ldots, 2M\}$. Then, the fact that $Q_k(\bar{d}_k)$ is bounded away from zero for $k \in K_{11}$ contradicts (18). Therefore, (19) is proved. □

Proof of Theorem 2.1. Let $\{x_k\}_{k \in K_0}$ an arbitrary convergent subsequence of $\{x_k\}_{k \in I\!N}$. By (19) we see that the hypothesis of Lemma 2.5 above holds with K_0 replacing K_1. Therefore, the limit of $\{x_k\}_{k \in K_0}$ is stationary, as we wanted to prove. □

Remark. We are especially interested in the spectral gradient choice of B_k. In this case,

$$B_k = \frac{1}{\lambda_k^{spg}} I$$

where

$$\lambda_k^{spg} = \begin{cases} \min(\lambda_{\max}, \max(\lambda_{\min}, s_k^T s_k / s_k^T y_k)), & \text{if } s_k^T y_k > 0, \\ \lambda_{\max}, & \text{otherwise,} \end{cases}$$

$s_k = x_k - x_{k-1}$ and $y_k = g_k - g_{k-1}$; so that

$$Q_k(d) = \frac{\|d\|^2}{2\lambda_k^{spg}} + g_k^T d. \tag{20}$$

3 Computing approximate projections

When $B_k = (1/\lambda_k^{spg})I$ (spectral choice) the optimal direction $\bar{d}_k$ is obtained by projecting $x_k - \lambda_k^{spg} g_k$ onto Ω, with respect to the Euclidean norm. Projecting onto Ω is a difficult problem unless Ω is an *easy* set (i.e. it is easy to project onto it) as a box, an affine subspace, a ball, etc. Fortunately, in many important applications, either Ω is an easy set or can be written as the intersection of a finite collection of closed and convex easy sets. In this work we are mainly concerned with extending the machinery developed in [6, 7] for the first case, to the second case. A suitable tool for this task is Dykstra's alternating projection algorithm, that will be described below. Dykstra's algorithm can also be obtained via duality [23] (see [12] for a complete discussion on this topic). Roughly speaking, Dykstra's algorithm projects in a clever way onto the easy convex sets individually to complete a cycle which is repeated iteratively. As an iterative method, it can be stopped prematurely to obtain d_k, instead of $\bar{d}_k$, such that $x_k + d_k \in \Omega$ and (4) holds. The fact that the process can be stopped prematurely could save significant computational work, and represents the inexactness of our algorithm.

Let us recall that for a given nonempty closed and convex set Ω of $\mathbb{R}^n$, and any $y^0 \in \mathbb{R}^n$, there exists a unique solution y^* to the problem

$$\min_{y \in \Omega} \|y^0 - y\|, \tag{21}$$

which is called the projection of y_0 onto Ω and is denoted by $P_\Omega(y_0)$. Consider the case $\Omega = \cap_{i=1}^p \Omega_i$, where, for $i = 1, \ldots, p$, Ω_i are closed and convex sets. Moreover, we assume that for all $y \in \mathbb{R}^n$, the calculation of $P_\Omega(y)$ is a difficult task, whereas, for each Ω_i, $P_{\Omega_i}(y)$ is easy to obtain.

Dykstra's algorithm [8, 13], solves (21) by generating two sequences, $\{y_i^\ell\}$ and $\{z_i^\ell\}$. These sequences are defined by the following recursive formulae:

$$\begin{aligned} y_0^\ell &= y_p^{\ell-1} \\ y_i^\ell &= P_{\Omega_i}(y_{i-1}^\ell - z_i^{\ell-1}) \quad , \quad i = 1, \ldots, p, \\ z_i^\ell &= y_i^\ell - (y_{i-1}^\ell - z_i^{\ell-1}) \quad , \quad i = 1, \ldots, p, \end{aligned} \tag{22}$$

for $\ell = 1, 2, \ldots$ with initial values $y_p^0 = y^0$ and $z_i^0 = 0$ for $i = 1, \ldots, p$.

Remarks

1. The increment $z_i^{\ell-1}$ associated with Ω_i in the previous cycle is always subtracted before projecting onto Ω_i. Therefore, only one increment (the last one) for each Ω_i needs to be stored.

2. If Ω_i is a closed affine subspace, then the operator P_{Ω_i} is linear and it is not necessary in the ℓ^{th} cycle to subtract the increment $z_i^{\ell-1}$ before projecting onto Ω_i. Thus, for affine subspaces, Dykstra's procedure reduces to the alternating projection method of von Neumann [30]. To be precise, in this case, $P_{\Omega_i}(z_i^{\ell-1}) = 0$.

3. For $\ell = 1, 2, \ldots$ and $i = 1, \ldots, p$, it is clear from (22) that the following relations hold

$$y_p^{\ell-1} - y_1^{\ell} = z_1^{\ell-1} - z_1^{\ell}, \tag{23}$$

$$y_{i-1}^{\ell} - y_i^{\ell} = z_i^{\ell-1} - z_i^{\ell}, \tag{24}$$

   where $y_p^0 = y^0$ and $z_i^0 = 0$, for all $i = 1, \ldots, p$.

For the sake of completeness we now present the key theorem associated with Dykstra's algorithm.

Theorem 3.1 (Boyle and Dykstra, 1986 [8]) *Let $\Omega_1, \ldots, \Omega_p$ be closed and convex sets of $\mathbb{R}^n$ such that $\Omega = \cap_{i=1}^p \Omega_i \neq \emptyset$. For any $i = 1, \ldots, p$ and any $y^0 \in \mathbb{R}^n$, the sequence $\{y_i^\ell\}$ generated by (22) converges to $y^* = P_\Omega(y^0)$ (i.e., $\|y_i^\ell - y^*\| \to 0$ as $\ell \to \infty$).*

A close inspection of the proof of the Boyle-Dykstra convergence theorem allows us to establish, in our next result, an interesting inequality that is suitable for the stopping process of our inexact algorithm.

Theorem 3.2 *Let y^0 be any element of $\mathbb{R}^n$ and define c_ℓ as*

$$c_\ell = \sum_{m=1}^{\ell} \sum_{i=1}^{p} \|y_{i-1}^m - y_i^m\|^2 + 2 \sum_{m=1}^{\ell-1} \sum_{i=1}^{p} \langle z_i^m, y_i^{m+1} - y_i^m \rangle. \tag{25}$$

Then, in the ℓ^{th} cycle of Dykstra's algorithm,

$$\|y^0 - y^*\|^2 \geq c_\ell \tag{26}$$

Moreover, at the limit when ℓ goes to infinity, equality is attained in (26).

Proof. In the proof of Theorem 3.1, the following equation is obtained [8, p. 34] (see also Lemma 9.19) in [12])

$$
\begin{aligned}
\|y^0 - y^*\|^2 = \|y_p^\ell - y^*\|^2 \quad &+ \quad \sum_{m=1}^{\ell} \sum_{i=1}^{p} \|z_i^{m-1} - z_i^m\|^2 \\
&+ \quad 2 \sum_{m=1}^{\ell-1} \sum_{i=1}^{p} \langle y_{i-1}^m - z_i^{m-1} - y_i^m, y_i^m - y_i^{m+1} \rangle \qquad (27) \\
&+ \quad 2 \sum_{i=1}^{p} \langle y_{i-1}^\ell - z_i^{\ell-1} - y_i^\ell, y_i^\ell - y^* \rangle,
\end{aligned}
$$

where all terms involved are nonnegative for all ℓ. Hence, we obtain

$$
\|y^0 - y^*\|^2 \geq \sum_{m=1}^{\ell} \sum_{i=1}^{p} \|z_i^{m-1} - z_i^m\|^2 + 2 \sum_{m=1}^{\ell-1} \sum_{i=1}^{p} \langle y_{i-1}^m - z_i^{m-1} - y_i^m, y_i^m - y_i^{m+1} \rangle. \qquad (28)
$$

Finally, (26) is obtained by replacing (23) and (24) in (28).

Clearly, in (27) all terms in the right hand side are bounded. In particular, using (23) and (24), the fourth term can be written as $2\sum_{i=1}^{p} \langle z_i^\ell, y_i^\ell - y^* \rangle$, and using the Cauchy-Schwarz inequality and Theorem 3.1, we notice that it vanishes when ℓ goes to infinity. Similarly, the first term in (27) tends to zero when ℓ goes to infinity, and so at the limit equality is attained in (26). □

Each iterate of the Dykstra's method is labeled by two indices i and ℓ. From now on we considered the subsequence with $i = p$ so that only one index ℓ is necessary. This will simplify considerably the notation without loss of generality. So, we assume that Dykstra's algorithm generates a single sequence $\{y^\ell\}$, so that

$$
y^0 = x_k - \lambda_k^{spg} g_k
$$

and

$$
\lim_{k \to \infty} y^\ell = y^* = P_\Omega(y^0).
$$

Moreover, by Theorem 3.2 we have that $\lim_{\ell \to \infty} c_\ell = \|y^0 - y^*\|^2$.

In the rest of this section we show how Dykstra's algorithm can be used to obtain a direction d_k that satisfies (4). First, we need a simple lemma related to convergence of sequences to points in convex sets whose interior is not empty.

Lemma 3.1 *Assume that Ω is a closed and convex set, $x \in Int(\Omega)$ and $\{y^\ell\} \subset I\!R^n$ is a sequence such that*

$$
\lim_{\ell \to \infty} y^\ell = y^* \in \Omega.
$$

For all $\ell \in I\!N$ we define

$$
\alpha_{\max}^\ell = \max\{\alpha \geq 0 \mid [x, x + \alpha(y^\ell - x)] \subset \Omega\} \qquad (29)
$$

and
$$x^\ell = x + \min(\alpha^\ell_{\max}, 1)(y^\ell - x). \tag{30}$$

Then,
$$\lim_{\ell\to\infty} x^\ell = y^*.$$

Proof. By (30), it is enough to prove that
$$\lim_{\ell\to\infty} \min(\alpha^\ell_{\max}, 1) = 1.$$

Assume that this is not true. Since $\min(\alpha^\ell_{\max}, 1) \leq 1$ there exists $\bar{\alpha} < 1$ such that for an infinite set of indices ℓ,
$$\min(\alpha^\ell_{\max}, 1) \leq \bar{\alpha}. \tag{31}$$

Now, by the convexity of Ω and the fact that x belongs to its interior, we have that
$$x + \frac{\bar{\alpha}+1}{2}(y^* - x) \in Int(\Omega).$$

But
$$\lim_{\ell\to\infty} x + \frac{\bar{\alpha}+1}{2}(y^\ell - x) = x + \frac{\bar{\alpha}+1}{2}(y^* - x),$$
then, for ℓ large enough
$$x + \frac{\bar{\alpha}+1}{2}(y^\ell - x) \in Int(\Omega).$$

This contradicts the fact that (31) holds for infinitely many indices. □

Lemma 3.2 *For all $z \in I\!R^n$,*
$$\|y^0 - z\|^2 = 2\lambda_k^{spg} Q_k(z - x_k) + \|\lambda_k^{spg} g_k\|^2. \tag{32}$$

Moreover,
$$\|y^0 - y^*\|^2 = 2\lambda_k^{spg} Q_k(\bar{d}_k) + \|\lambda_k^{spg} g_k\|^2. \tag{33}$$

Proof.
$$\begin{aligned}
\|y^0 - z\|^2 &= \|x_k - \lambda_k^{spg} g_k - z\|^2 \\
&= \|x_k - z\|^2 - 2\lambda_k^{spg}(x_k - z)^T g_k + \|\lambda_k^{spg} g_k\|^2 \\
&= 2\lambda_k^{spg}[\frac{\|z - x_k\|^2}{2\lambda_k^{spg}} + (z - x_k)^T g_k] + \|\lambda_k^{spg} g_k\|^2 \\
&= 2\lambda_k^{spg} Q_k(z - x_k) + \|\lambda_k^{spg} g_k\|^2
\end{aligned}$$

Therefore, (32) is proved. By this identity, if y^* is the minimizer of $\|y^0 - z\|^2$ for $z \in \Omega$, then $y^* - x_k$ must be the minimizer of $Q_k(d)$ for $x_k + d \in \Omega$. Therefore,
$$y^* = x_k + \bar{d}_k.$$

So, (33) also holds. □

Lemma 3.3 *For all $\ell \in I\!N$, define*

$$a_\ell = \frac{c_\ell - \|\lambda_k^{spg} g_k\|^2}{2\lambda_k^{spg}}. \tag{34}$$

Then

$$a_\ell \leq Q_k(\bar{d}_k) \ \forall \ell \in I\!N \tag{35}$$

and

$$\lim_{\ell \to \infty} a_\ell = Q_k(\bar{d}_k).$$

Proof. By Lemma 3.2,

$$Q_k(z - x_k) = \frac{\|y^0 - z\|^2 - \|\lambda_k^{spg} g_k\|^2}{2\lambda_k^{spg}}.$$

By (26), $\|y^\ell - z\|^2 \geq c_\ell$ for all $z \in \Omega$ and for all $\ell \in I\!N$. Therefore, for all $z \in \Omega$, $\ell \in I\!N$,

$$Q_k(z - x_k) \geq \frac{c_\ell - \|\lambda_k^{spg} g_k\|^2}{2\lambda_k^{spg}} = a_\ell.$$

In particular, if $z - x_k = \bar{d}_k$, we obtain (35). Moreover, since $\lim_{\ell \to \infty} c_\ell = \|y^0 - y^*\|^2$, we have that

$$\lim_{\ell \to \infty} a_\ell = Q_k(y^* - x_k) = Q_k(\bar{d}_k).$$

This completes the proof. □

By the three lemmas above, we have established that, using Dykstra's algorithm we are able to compute a sequence $\{x_k^\ell\}_{\ell \in I\!N}$ such that

$$x_k^\ell \in \Omega \ \forall \ell \in I\!N \text{ and } x_k^\ell - x_k \to \bar{d}_k$$

and, consequently,

$$Q_k(x_k^\ell - x_k) \to Q_k(\bar{d}_k). \tag{36}$$

Moreover, we proved that $a_\ell \leq Q_k(\bar{d}_k)$ for all $\ell \in I\!N$ and that

$$\lim_{\ell \to \infty} a_\ell = Q_k(\bar{d}_k). \tag{37}$$

Since $x_k^\ell \in \Omega$ we also have that $Q_k(x_k^\ell - x_k) \geq Q_k(\bar{d}_k)$ for all $\ell \in I\!N$.

If x_k is not stationary (so $Q_k(\bar{d}_k) < 0$), given an arbitrary $\eta' \in (\eta, 1)$, the properties (36) and (37) guarantee that, for ℓ large enough,

$$Q_k(x_k^\ell - x_k) \leq \eta' a_\ell. \tag{38}$$

So,

$$Q_k(x_k^\ell - x_k) \leq \eta' Q_k(\bar{d}_k). \tag{39}$$

The inequality (38) can be tested at each iteration of the Dykstra's algorithm. When it holds, we obtain x_k^ℓ satisfying (39).

The success of this procedure depends on the fact of x_k being interior. The point x_k^ℓ so far obtained belongs to Ω but is not necessarily interior. A measure of the "interiority" of x_k^ℓ can be given by $\alpha_{\max}^\ell$ (defined by (29)). Define $\beta = \eta/\eta'$. If $\alpha_{\max}^\ell \geq 1/\beta$, the point x_k^ℓ is considered to be safely interior. If $\alpha_{\max}^\ell \leq 1/\beta$, the point x_k^ℓ may be interior but excessively close to the boundary or even on the boundary (if $\alpha_{\max}^\ell \leq 1$). Therefore, the direction d_k is taken as

$$d_k = \begin{cases} (x_k^\ell - x_k), & \text{if } \alpha_{\max}^\ell \in [1/\beta, \infty), \\ \beta\, \alpha_{\max}^\ell\, (x_k^\ell - x_k), & \text{if } \alpha_{\max}^\ell \in [1, 1/\beta], \\ \beta\, (x_k^\ell - x_k), & \text{if } \alpha_{\max}^\ell \in [0, 1]. \end{cases} \tag{40}$$

Note that $d_k = \omega(x_k^\ell - x_k)$ with $\omega \in [\beta, 1]$. In this way, $x_k + d_k \in Int(\Omega)$ and, by the convexity of Q_k,

$$Q_k(d_k) = Q_k(\omega(x_k^\ell - x_k)) \leq \omega Q_k(x_k^\ell - x_k) \leq \beta\eta' Q_k(\bar{d}_k) = \eta Q_k(\bar{d}_k).$$

Therefore, the vector d_k obtained in (40) satisfies (4). Observe that the "reduction" (40) is performed only once, at the end of the Dykstra's process, when (39) has already been satisfied. Moreover, by (29) and (30), definition (40) is equivalent to

$$d_k = \begin{cases} (y^\ell - x_k), & \text{if } \alpha_{\max}^\ell \geq 1/\beta, \\ \beta\, \alpha_{\max}^\ell\, (y^\ell - x_k), & \text{otherwise.} \end{cases}$$

The following algorithm condenses the procedure described above for computing a direction that satisfies (4).

Algorithm 3.1: Compute approximate projection

Assume that $\varepsilon > 0$ (small), $\beta \in (0,1)$ and $\eta' \in (0,1)$ are given ($\eta \equiv \beta\, \eta'$).

Step 1.

Set $\ell \leftarrow 0$, $y^0 = x_k - \lambda_k^{spg} g_k$, $c_0 = 0$, $a_0 = -\|\lambda_k^{spg} g_k\|^2$, and compute x_k^0 by (30) for $x = x_k$.

Step 2.

If (38) is satisfied, compute d_k by (40) and terminate the execution of the algorithm. The approximate projection has been successfully computed.

If $-a_\ell \leq \varepsilon$, stop. Probably, a point satisfying (38) does not exist.

Step 3.

Compute $y^{\ell+1}$ using Dykstra's algorithm (22), $c_{\ell+1}$ by (25), $a_{\ell+1}$ by (34), and $x_k^{\ell+1}$ by (30) for $x = x_k$.

Step 4.
Set $\ell \leftarrow \ell + 1$ and go to Step 2.

The results in this section show that Algorithm 3.1 stops giving a direction that satisfies (4) whenever $Q_k(\bar{d}_k) < 0$. The case $Q_k(\bar{d}_k) = 0$ is possible, and corresponds to the case in which x_k is stationary. Accordingly, a criterion for stopping the algorithm when $Q_k(\bar{d}_k) \approx 0$ has been incorporated. The lower bound a_ℓ allows one to establish such criterion. Since $a_\ell \leq Q_k(\bar{d}_k)$ and $a_\ell \rightarrow Q_k(\bar{d}_k)$ the algorithm is stopped when $-a_\ell \leq \varepsilon$ where $\varepsilon > 0$ is a small tolerance given by the user. When this happens, the point x_k can be considered nearly stationary for the original problem.

4 Numerical Results

4.1 Test problem

Interesting applications appear as constrained least-squares rectangular matrix problems. In particular, we consider the following problem:

$$\begin{array}{ll} \text{Minimize} & \|AX - B\|_F^2 \\ \text{subject to} & \\ & X \in SDD^+ \\ & 0 \leq L \leq X \leq U, \end{array} \tag{41}$$

where A and B are given $nrows \times ncols$ real matrices, $nrows \geq ncols$, $rank(A) = ncols$, and X is the symmetric $ncols \times ncols$ matrix that we wish to find. For the feasible region, L and U are given $ncols \times ncols$ real matrices, and SDD^+ represents the cone of symmetric and diagonally dominant matrices with positive diagonal, i.e.,

$$SDD^+ = \{X \in I\!R^{ncols \times ncols} \mid X^T = X \text{ and } x_{ii} \geq \sum_{j \neq i} |x_{ij}| \text{ for all } i\}.$$

Throughout this section, the notation $A \leq B$, for any two real $ncols \times ncols$ matrices, means that $A_{ij} \leq B_{ij}$ for all $1 \leq i, j \leq ncols$. Also, $\|A\|_F$ denotes the Frobenius norm of a real matrix A, defined as

$$\|A\|_F^2 = \langle A, A \rangle = \sum_{i,j} (a_{ij})^2 \ ,$$

where the inner product is given by $\langle A, B \rangle = trace(A^T B)$. In this inner product space, the set S of symmetric matrices form a closed subspace and SDD^+ is a closed and convex polyhedral cone [1, 18, 16]. Therefore, the feasible region is a closed and convex set in $I\!R^{ncols \times ncols}$.

Problems closely related to (41) arise naturally in statistics and mathematical economics [13, 14, 17, 24]. An effective way of solving (41) is by means of alternating projection methods combined with a geometrical understanding of the feasible region. For the simplified case in which $nrows = ncols$, A is the identity matrix, and the bounds are not taken into account, the problem has been solved in [26, 29] using Dykstra's alternating projection algorithm. Under this approach, the symmetry and the sparsity pattern of the given matrix B are preserved, and so it is of interest for some numerical optimization techniques discussed in [26].

Unfortunately, the only known approach for using alternating projection methods on the general problem (41) is based on the use of the singular value decomposition (SVD) of the matrix A (see for instance [15]), and this could lead to a prohibitive amount of computational work in the large scale case. However, problem (41) can be viewed as a particular case of (1), in which $f : I\!R^{ncols\times(ncols+1)/2} \to I\!R$, is given by

$$f(X) = \|AX - B\|_F^2,$$

and $\Omega = Box \bigcap SDD^+$, where $Box = \{X \in I\!R^{ncols\times ncols} \mid L \le X \le U\}$. Hence, it can be solved by means of the ISPG algorithm. Notice that, since $X = X^T$, the function f is defined on the subspace of symmetric matrices. Notice also that, instead of expensive factorizations, it is now required to evaluate the gradient matrix, given by

$$\nabla f(X) = 2A^T(AX - B).$$

In order to use the ISPG algorithm, we need to project inexactly onto the feasible region. For that, we make use of Dykstra's alternating projection method. For computing the projection onto SDD^+ we make use of the procedure developed introduced in [29].

4.2 Implementation details

We implemented Algorithm 2.1 with the definition (20) of Q_k and Algorithm 3.1 for computing the approximate projections.

The unknowns of our test problem are the $n = ncols \times (ncols+1)/2$ entries of the upper triangular part of the symmetric matrix X. The projection of X onto SDD^+ consists on several cycles of projections onto the $ncols$ convex sets

$$SDD_i^+ = \{X \in I\!R^{ncols\times ncols} \mid x_{ii} \ge \sum_{j\ne i} |x_{ij}|\}$$

(see [29] for details). Since projecting onto SDD_i^+ only involves the row/column i of X, then all the increments $z_i^{\ell-1}$ can be saved in a unique vector $v^{\ell-1} \in I\!R^n$, which is consistent with the low memory requirements of the SPG-like methods.

We use the convergence criteria given by

$$\|x_k^\ell - x_k\|_\infty \le \epsilon_1 \text{ or } \|x_k^\ell - x_k\|_2 \le \epsilon_2,$$

where x_k^ℓ is the iterate of Algorithm 3.1 which satisfies inequality (38).

The arbitrary initial spectral steplength $\lambda_0 \in [\lambda_{\min}, \lambda_{\max}]$ is computed as

$$\lambda_0^{spg} = \begin{cases} \min(\lambda_{\max}, \max(\lambda_{\min}, \bar{s}^T\bar{s}/\bar{s}^T\bar{y})), & \text{if } \bar{s}^T\bar{y} > 0, \\ \lambda_{\max}, & \text{otherwise,} \end{cases}$$

where $\bar{s} = \bar{x} - x_0$, $\bar{y} = g(\bar{x}) - g(x_0)$, $\bar{x} = x_0 - t_{small}\nabla f(x_0)$,$t_{small}$ is a small number defined as $t_{small} = \max(\epsilon_{rel}\|x\|_\infty, \epsilon_{abs})$ with ϵ_{rel} a relative small number and ϵ_{abs} an absolute small number.

The computation of α_{new} uses one-dimensional quadratic interpolation and it is safeguarded taking $\alpha_{\text{new}} \leftarrow \alpha/2$ when the minimum of the one-dimensional quadratic lies outside $[\sigma_1, \sigma_2\alpha]$.

In the experiments, we chose $\epsilon_1 = \epsilon_2 = 10^{-5}$, $\epsilon_{rel} = 10^{-7}$, $\epsilon_{abs} = 10^{-10}$, $\beta = 0.85$, $\gamma = 10^{-4}$, $\sigma_1 = 0.1$, $\sigma_2 = 0.9$, $\lambda_{\min} = 10^{-3}$, $\lambda_{\max} = 10^3$, $M = 10$. Different runnings were made with $\eta' = 0.7$, 0.8, 0.9 and 0.99 ($\eta = \beta\eta' = 0.595$, 0.68, 0.765 and 0.8415, respectively) to compare the influence of the inexact projections in the overall performance of the method.

4.3 Experiments

All the experiments were run on a Sun Ultra 60 Workstation with 2 UltraSPARC-II processors at 296-Mhz, 512 Mb of main memory, and SunOS 5.7 operating system. The compiler was Sun WorkShop Compiler Fortran 77 4.0 with flag -O to optimize the code.

We generated a set of 10 random matrices of dimensions 10×10 up to 100×100. The matrices A, B, and the initial guess X_0 are randomly generated, with elements in the interval $[0, 1]$. We use the Schrage's random number generator [31] (double precision version) with seed equal to 1 for a machine-independent generation of random numbers. Matrix X_0 is then redefined as $(X_0 + X_0^T)/2$ and, its diagonal elements A_{ii} are again redefined as $2\ \sum_{j\neq i} |A_{ij}|$ to guarantee an interior feasible initial guess. Bounds L and U are defined as $L \equiv 0$ and $U \equiv \infty$.

Tables 1–4 display the performance of ISPG with $\eta' = 0.7$, 0.8, 0.9 and 0.99, respectively. The columns mean: n, dimension of the problem; IT, iterations needed to reach the solution; FE, function evaluations; GE, gradient evaluations, DIT, Dykstra's iterations; *Time*, CPU time (seconds); f, function value at the solution; $\|d\|_\infty$, sup-norm of the ISPG direction; and $\alpha_{\max}$, maximum feasible step on that direction. Observe that, as expected, α_{max} is close to 1 when we solve the quadratic subproblem with high precision ($\eta' \approx 1$).

In all the cases we found the same solutions. These solutions were never interior points. When we compute the projections with high precision the number of outer iterations decreases. Of course, in that case, the cost of computing an approximate projection using Dykstra's algorithm increases. Therefore, optimal efficiency of the algorithm comes from a compromise between those two tendencies. The best value for η' seems to be 0.8 in this set of experiments.

5 Final remarks

We present a new algorithm for convex constrained optimization. At each iteration, a search direction is computed as an approximate solution of a quadratic subproblem and, in the implementation, the set of iterates are interior. We prove global convergence, using a nonmonotone line search procedure of the type introduced in [22] and used in several papers since then.

n	IT	FE	GE	DIT	Time	f	$\|d\|_\infty$	$\alpha_{\max}$
100	28	29	30	1139	0.48	2.929D+01	1.491D-05	8.566D-01
400	34	35	36	693	2.20	1.173D+02	4.130D-06	4.726D-01
900	23	24	25	615	8.03	2.770D+02	1.450D-05	1.000D+00
1600	23	24	25	808	25.16	5.108D+02	8.270D-06	7.525D-01
2500	22	23	24	473	36.07	7.962D+02	1.743D-05	7.390D-01
3600	22	23	24	513	56.75	1.170D+03	8.556D-06	7.714D-01
4900	20	21	22	399	78.39	1.616D+03	1.888D-05	7.668D-01
6400	21	22	23	523	153.77	2.133D+03	1.809D-05	7.989D-01
8100	21	22	23	610	231.07	2.664D+03	1.197D-05	7.322D-01
10000	21	22	23	541	283.07	3.238D+03	1.055D-05	7.329D-01

Table 1: ISPG performance with inexactness parameter $\eta' = 0.7$.

n	IT	FE	GE	DIT	Time	f	$\|d\|_\infty$	$\alpha_{\max}$
100	25	26	27	1012	0.43	2.929D+01	1.252D-05	1.000D+00
400	30	31	32	579	1.81	1.173D+02	1.025D-04	1.000D+00
900	22	23	24	561	6.96	2.770D+02	2.045D-05	9.623D-01
1600	21	22	23	690	20.93	5.108D+02	1.403D-05	8.197D-01
2500	21	22	23	575	35.58	7.962D+02	1.087D-05	8.006D-01
3600	20	21	22	409	43.33	1.170D+03	1.485D-05	8.382D-01
4900	19	20	21	496	83.61	1.616D+03	1.683D-05	8.199D-01
6400	18	19	20	465	121.20	2.133D+03	1.356D-05	9.123D-01
8100	18	19	20	451	168.87	2.664D+03	2.333D-05	8.039D-01
10000	19	20	21	498	261.58	3.238D+03	1.209D-05	8.163D-01

Table 2: ISPG performance with inexactness parameter $\eta' = 0.8$.

n	IT	FE	GE	DIT	Time	f	$\|d\|_\infty$	$\alpha_{\max}$
100	26	27	28	1363	0.59	2.929D+01	1.195D-05	9.942D-01
400	26	27	28	512	1.76	1.173D+02	2.981D-04	2.586D-02
900	21	22	23	527	6.62	2.770D+02	1.448D-05	9.428D-01
1600	21	22	23	886	28.64	5.108D+02	1.441D-05	9.256D-01
2500	20	21	22	537	37.21	7.962D+02	1.559D-05	9.269D-01
3600	20	21	22	518	54.69	1.170D+03	1.169D-05	9.122D-01
4900	18	19	20	509	87.27	1.616D+03	2.169D-05	9.080D-01
6400	17	18	19	557	148.49	2.133D+03	1.366D-05	9.911D-01
8100	17	18	19	510	198.69	2.664D+03	1.968D-05	9.051D-01
10000	18	19	20	585	323.51	3.238D+03	1.557D-05	9.154D-01

Table 3: ISPG performance with inexactness parameter $\eta' = 0.9$.

n	IT	FE	GE	DIT	Time	f	$\|d\|_\infty$	$\alpha_{\max}$
100	21	22	23	1028	0.46	2.929D+01	2.566D-05	8.216D-01
400	25	26	27	596	1.95	1.173D+02	5.978D-05	2.682D-01
900	20	21	22	715	8.69	2.770D+02	9.671D-06	9.796D-01
1600	19	20	21	1037	31.24	5.108D+02	1.538D-05	9.890D-01
2500	19	20	21	827	50.07	7.962D+02	1.280D-05	9.904D-01
3600	17	18	19	654	69.40	1.170D+03	1.883D-05	9.911D-01
4900	17	18	19	805	153.57	1.616D+03	2.337D-05	9.926D-01
6400	16	17	18	828	229.72	2.133D+03	1.163D-05	9.999D-01
8100	16	17	18	763	312.84	2.664D+03	2.536D-05	9.924D-01
10000	16	17	18	660	403.82	3.238D+03	1.795D-05	9.920D-01

Table 4: ISPG performance with inexactness parameter $\eta' = 0.99$.

A particular case of the model algorithm is the inexact spectral projected gradient method (ISPG) which turns out to be a generalization of the spectral projected gradient (SPG) method introduced in [6, 7]. The ISPG must be used instead of SPG when projections onto the feasible set are not easy to compute. In the present implementation we use Dykstra's algorithm [13] for computing approximate projections. If, in the future, acceleration techniques are developed for Dykstra's algorithm, they can be included in the ISPG machinery (see [12, pp.235]).

Numerical experiments were presented concerning constrained least-squares rectangular matrix problems to illustrate the good features of the ISPG method.

Acknowledgements.

We are indebted to the associate editor and an anonymous referee whose comments helped us to improve the final version of this paper.

References

[1] G. P. Barker and D. Carlson [1975], Cones of diagonally dominant matrices, *Pacific Journal of Mathematics* 57, pp. 15-32.

[2] J. Barzilai and J. M. Borwein [1988], Two point step size gradient methods, *IMA Journal of Numerical Analysis* 8, pp. 141–148.

[3] D. P. Bertsekas [1976], On the Goldstein-Levitin-Polyak gradient projection method, *IEEE Transactions on Automatic Control* 21, pp. 174–184.

[4] D. P. Bertsekas [1999], *Nonlinear Programming*, Athena Scientific, Belmont, MA.

[5] E. G. Birgin and J. M. Martínez [2002], Large-scale active-set box-constrained optimization method with spectral projected gradients, *Computational Optimization and Applications* 23, pp. 101–125.

[6] E. G. Birgin, J. M. Martínez and M. Raydan [2000], Nonmonotone spectral projected gradient methods on convex sets, *SIAM Journal on Optimization* 10, pp. 1196-1211.

[7] E. G. Birgin, J. M. Martínez and M. Raydan [2001], Algorithm 813: SPG - Software for convex-constrained optimization, *ACM Transactions on Mathematical Software* 27, pp. 340-349.

[8] J. P. Boyle and R. L. Dykstra [1986], A method for finding projections onto the intersection of convex sets in Hilbert spaces, *Lecture Notes in Statistics* 37, pp. 28–47.

[9] Y. H. Dai and L. Z. Liao [2002], R-linear convergence of the Barzilai and Borwein gradient method, *IMA Journal on Numerical Analysis* 22, pp. 1–10.

[10] Y. H. Dai [2000], On nonmonotone line search, *Journal of Optimization Theory and Applications* , to appear.

[11] G. B. Dantzig, *Deriving an utility function for the economy*, SOL 85-6R, Department of Operations Research, Stanford University, CA, 1985.

[12] F. Deutsch [2001], *Best Approximation in Inner Product Spaces*, Springer Verlag New York, Inc.

[13] R. L. Dykstra [1983], An algorithm for restricted least-squares regression, *Journal of the American Statistical Association* 78, pp. 837–842.

[14] R. Escalante and M. Raydan [1996], Dykstra's Algorithm for a Constrained Least-Squares Matrix Problem, *Numerical Linear Algebra and Applications* 3, pp. 459-471.

[15] R. Escalante and M. Raydan [1998], On Dykstra's algorithm for constrained least-squares rectangular matrix problems, *Computers and Mathematics with Applications* 35, pp. 73-79.

[16] M. Fiedler and V. Ptak [1967], Diagonally dominant matrices, *Czech. Math. J.* 17, pp. 420-433.

[17] R. Fletcher [1981], A nonlinear programming problem in statistics (educational testing), *SIAM Journal on Scientific and Statistical Computing* 2, pp. 257–267.

[18] R. Fletcher [1985], Semi-definite matrix constraints in optimization, *SIAM Journal on Control and Optimization* 23, pp. 493–513.

[19] R. Fletcher [1990], Low storage methods for unconstrained optimization, *Lectures in Applied Mathematics (AMS)* 26, pp. 165–179.

[20] R. Fletcher [2001], On the Barzilai-Borwein method, *Department of Mathematics, University of Dundee* NA/207, Dundee, Scotland.

[21] A. A. Goldstein [1964], Convex Programming in Hilbert Space, *Bulletin of the American Mathematical Society* 70, pp. 709–710.

[22] L. Grippo, F. Lampariello and S. Lucidi [1986], A nonmonotone line search technique for Newton's method, *SIAM Journal on Numerical Analysis* 23, pp. 707-716.

[23] S. P. Han [1988], A successive projection method, *Mathematical Programming* 40, pp. 1–14.

[24] H. Hu and I. Olkin [1991], A numerical procedure for finding the positive definite matrix closest to a patterned matrix, *Statistics and Probability letters* 12, pp. 511-515.

[25] E. S. Levitin and B. T. Polyak [1966], Constrained Minimization Problems, *USSR Computational Mathematics and Mathematical Physics* 6, pp. 1–50.

[26] M. Mendoza, M. Raydan and P. Tarazaga [1998], Computing the nearest diagonally dominant matrix, *Numerical Linear Algebra with Applications* 5, pp. 461-474.

[27] M. Raydan [1993], On the Barzilai and Borwein choice of steplength for the gradient method, *IMA Journal of Numerical Analysis* 13, pp. 321–326.

[28] M. Raydan [1997], The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, *SIAM Journal on Optimization* 7, pp. 26–33.

[29] M. Raydan and P. Tarazaga [2002], Primal and polar approach for computing the symmetric diagonally dominant projection, *Numerical Linear Algebra with Applications* 9, pp. 333-345.

[30] J. von Neumann [1950], Functional operators vol. II. The geometry of orthogonal spaces, *Annals of Mathematical Studies* 22, Princeton University Press. This is a reprint of mimeographed lecture notes first distributed in 1933.

[31] L. Schrage, A more portable Fortran random number generator, *ACM Transactions on Mathematical Software* 5, pp. 132–138 (1979).