Paper 34

Proceedings of the Seventh International Conference on Engineering Computational Technology, B.H.V. Topping, J.M. Adam, F.J. Pallarés, R. Bru and M.L. Romero, (Editors), Civil-Comp Press, Stirlingshire, Scotland

Inexact Jacobian Constraint Preconditioners in Optimization

L. Bergamaschi[1], M. Venturin[2] and G. Zilli[1]
[1] Department of Mathematical Methods and Models for Scientific Applications University of Padua, Italy
[2]Department of Computer Science University of Verona, Italy

Abstract

In this paper we analyze a class of approximate constraint preconditioners in the acceleration of Krylov subspace methods fot the solution of reduced Newton systems arising in optimization with interior point methods. We propose a dynamic sparsification of the Jacobian matrix at every stage of the interior point method. Spectral analysis of the preconditioned matrix is performed and bounds on its non-unit eigenvalues are provided. Preliminary computational results are encouraging.

Keywords: interior-point methods, iterative solvers, preconditioners, approximate Jacobian.

1 Introduction

In this paper we are concerned with the solution of large scale minimization problems subject to equality constraints, via interior point methods with iterative solvers. When minimization problems subject to equality constraints are considered, each iteration k of an interior point method requires the following linear system of equations to be solved

$$H_k x = b \qquad \text{where} \qquad H_k = \begin{bmatrix} Q_k & A^T \\ A & \end{bmatrix}, \tag{1}$$

where $Q_k \in \mathcal{R}^{n\times n}$ is the Hessian of Lagrangian and $A \in \mathcal{R}^{m\times n}$ is the Jacobian of constraints. The matrix Q_k arising in interior point applications has form $Q_k = Q_0 + \Theta_k$, where diagonal scaling matrix $\Theta_k \in \mathcal{R}^{n\times n}$ with strictly positive elements (due to the barrier terms for primal variables).

There has been growing interest in recent years in the use of iterative methods to solve system (1) arising in optimization context. This is because certain large instances of (1) defy direct methods (the inverse representation of the matrix involved requires prohibitive memory resources and cannot be computed efficiently). A variety of preconditioners have been proposed for such matrices, notably [3, 7, 8, 9] to mention a few. They have a common feature of constructing the approximation to (1) by simplifying its upper left block, namely by applying the preconditioner of the form

$$P_k = \begin{bmatrix} D_k & A^T \\ A & \end{bmatrix}. \tag{2}$$

Usual choices for matrix D_k are to keep it block-diagonal or diagonal. The latter situation has been studied for example in [3]: this choice has an advantage because it allows for further reduction of the preconditioner P to the form of normal equations (Schur complement) where a reduced system of form $AD^{-1}A^T$ is computed. The Hessian matrix Q cannot be approximated by anything simpler than a diagonal matrix. Consequently, the factorization of P (with a diagonal D) determines the least expensive constraint preconditioner among those preconditioners which "respect" constraints of the optimization problem. However, in certain situations such a preconditioner is still too expensive to compute.

In [2, 4] the following form of the preconditioner is proposed:

$$\tilde{P}_k = \begin{bmatrix} D_k & \tilde{A}^T \\ \tilde{A} & \end{bmatrix}, \tag{3}$$

where $D_k = \text{diag}(Q_k)$ and $\tilde{A}$ is a suitable approximation of the Jacobian matrix A Here we propose to compute dinamically the approximation to the Jacobian, namely we use

$$\tilde{P}_k = \begin{bmatrix} D_k & \tilde{A}_k^T \\ \tilde{A}_k & \end{bmatrix}, \tag{4}$$

i. e. we drop dynamically the elements from the ith row of A_k^T depending on the magnitude of the corresponding element of diagonal matrix D_k.

Dropping some of the elements in the Jacobian matrix A produces a significant reduction of the fill-in of the Cholesky factor of the preconditioner, thus speeding-up the cost of a single iteration of the Krylov subspace method of choice.

A spectral analysis is performed, which show that the eigenvalues of the preconditioned matrix are bounded in terms of the norm of two matrices that measure the error introduced by approximating Q_k with D_k and A by $\tilde{A}_k$.

Some numerical results onto a number of large quadratic problems demonstrate that the new approach is an attractive alternative to direct approach and to exact constraint preconditioners. The paper is organised as follows. In Section 2 we provide the spectral analysis of the preconditioner. In Section 3 we illustrate the behaviour of the preconditioner on a class of quadratic programming problems. In Section 4 is presented the dynamic dropping of the Jacobian matrix adopted. Finally, in Section 5 we give our conclusions.

2 Spectral Analysis

Following [2] we define $E = A - \tilde{A}$, $rank(E) = p$ and denote as $\tilde{\sigma}_1$ the smallest singular value of $\tilde{A}D^{-1/2}$. Further we introduce two error terms:

$$e_Q = \|E_Q\| = \|D^{-1/2}QD^{-1/2} - I\| \qquad \text{and} \qquad e_J = \frac{\|ED^{-1/2}\|}{\sigma_1(\tilde{A}D^{-1/2})} \tag{5}$$

which measure the errors of Hessian and Jacobian approximations, respectively. The distance of the complex eigenvalues from one ($\tau = \lambda - 1$) will be bounded in terms of these two quantities.

Theorem. Assume A and $\tilde{A}$ have maximum rank. If the eigenvector is of the form $(0, y)^T$ then the eigenvalues of $\tilde{P}^{-1}H$ are either one (with multiplicity at least $m - p$) or possibly complex and bounded by $|\tau| \leq e_J$. Corresponding to eigenvectors of the form $(x, y)^T$ with $x \neq 0$ the eigenvalues are

1. equal to one (with multiplicity at least $m - p$), or

2. real positive and bounded by

$$\lambda_{\min}(D^{-1}Q) \leq \lambda \leq \lambda_{\max}(D^{-1}Q), \; or$$

3. complex, satisfying

$$\begin{aligned} |\tau_R| &\leq e_Q + e_J \qquad (6) \\ |\tau_I| &\leq e_Q + e_J, \qquad (7) \end{aligned}$$

where $\tau = \tau_R + i\tau_I$.

Proof. The eigenvalues of $\tilde{P}^{-1}H$ are the same as those of $\bar{P}^{-1}\bar{H}$ where $\bar{P} = \mathcal{D}\tilde{P}\mathcal{D}$ and $\bar{H} = \mathcal{D}H\mathcal{D}$ and

$$\mathcal{D} = \begin{bmatrix} D^{-1/2} & 0 \\ 0 & I \end{bmatrix}$$

They must satisfy

$$\begin{cases} Ku + B^T y &= \lambda u + \lambda B^T y - \lambda F^T y \\ Bu &= \lambda Bu - \lambda Fu \end{cases} . \tag{8}$$

where $K = D^{-1/2}QD^{-1/2}, \; B = AD^{-1/2}, \; F = ED^{-1/2}, \; u = D^{1/2}x$. The eigenvalue problem can also be stated, setting $\tilde{B} = (A - E)D^{-1/2} = B - F$ and $\tau = \lambda - 1$,

$$\begin{cases} \tau u + \tau \tilde{B}^T y &= (K - I)u + F^T y \\ \tau \tilde{B} u &= Fu \end{cases} . \tag{9}$$

Let us observe that $K - I$ and F are the errors of approximation of the Hessian and Jacobian in (3), respectively.

We now analyse a number of cases depending on u and y.

1. $\boxed{u = 0}$ Every vector of the form $\begin{pmatrix} 0 \\ y \end{pmatrix}$ where y is an eigenvector of the generalized eigenproblem

$$BB^T y = \lambda B\tilde{B}^T y$$

is the eigenvector of (8) corresponding to λ. Since $rank(E) = p$ (hence $rank(F) = p$), among those vectors y there are $m - p$ satisfying $F^T y = 0$. The first equation of (8) reads

$$B^T y = \lambda B^T y$$

so that eigenvector $\begin{pmatrix} 0 \\ y \end{pmatrix}$ is associated to the unit eigenvalue.

We can bound the remaining such eigenvalues in terms of $\|F\|$ using again the first equation of (9) as

$$|\tau| = \frac{\|F^T y\|}{\|\tilde{B}^T y\|} \le \|F\| \frac{\|y\|}{\|\tilde{B}^T y\|} \le \frac{\|F\|}{\tilde{\sigma}_1} = e_J \,. \tag{10}$$

2. $\boxed{u \neq 0}$ There are at least $m - p$ linearly independent vectors u satisfying $Fu = 0$ and $Bu \neq 0$. For such vectors the second equation of (8) reads

$$Bu = \lambda Bu$$

giving again unit eigenvalues.

Let us consider now the most general case where $Bu \neq 0$ and $Fu \neq 0$.

Let us multiply the first equation of (8) by u^H and the second one by y^H. We obtain the following system

$$\begin{cases} u^H K u + u^H B^T y &= \lambda u^H u + \lambda u^H \tilde{B}^T y \\ y^H B u &= \lambda y^H \tilde{B} u \end{cases} . \tag{11}$$

We shall consider two possibilities: (a) real eigenvalues and (b) complex eigenvalues.

(a) $\boxed{\text{real eigenvalues}}$ If $\lambda \in \mathbb{R}$, subtracting the transpose of the second equation from the first one, and using $\lambda = \bar{\lambda}$, $(u, y)^H = (u, y)^T$, we obtain

$$\lambda_{\min}(K) \le \lambda = \frac{u^T K u}{u^T u} \le \lambda_{\max}(K). \tag{12}$$

(b) $\boxed{\text{complex eigenvalues}}$ To bound the complex eigenvalues we write in short equation (9) as

$$\tau N \begin{pmatrix} u \\ y \end{pmatrix} = M \begin{pmatrix} u \\ y \end{pmatrix}$$

and observe that matrix N^{-1} can be decomposed using the Cholesky factorization LL^T of the symmetric positive definite matrix $\tilde{B}\tilde{B}^T$.

$$N^{-1} = \begin{pmatrix} I & \tilde{B}^T \\ \tilde{B} & 0 \end{pmatrix}^{-1} = \begin{pmatrix} I & -\tilde{B}^T L^{-T} \\ 0 & L^{-T} \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix} \begin{pmatrix} I & 0 \\ -L^{-1}\tilde{B} & L^{-1} \end{pmatrix} = UJU^T \tag{13}$$

so that the eigenvalue problem is equivalent to

$$U^T M U w = \tau J w, \quad \text{with} \quad \begin{pmatrix} u \\ y \end{pmatrix} = Uw. \tag{14}$$

It is useful to set $R = L^{-1}\tilde{B}$ since it is easily found that $\|R\| = 1$. Since

$$\begin{aligned} U^T M U &= \begin{pmatrix} I & 0 \\ -R & L^{-1} \end{pmatrix} \begin{pmatrix} E_Q & F^T \\ F & 0 \end{pmatrix} \begin{pmatrix} I & -R^T \\ 0 & L^{-T} \end{pmatrix} \\ &= \begin{pmatrix} E_Q & F^T \\ -RE_Q + L^{-1}F & -RF^T \end{pmatrix} \begin{pmatrix} I & -R^T \\ 0 & L^{-T} \end{pmatrix} \\ &= \begin{pmatrix} E_Q & -E_Q R^T + F^T L^{-T} \\ -RE_Q + L^{-1}F & RE_Q R^T - L^{-1}FR^T - RF^T L^{-T} \end{pmatrix} \end{aligned}$$

we rewrite equation (14)

$$\begin{pmatrix} E_Q & -E_Q R^T + F^T L^{-T} \\ -RE_Q + L^{-1}F & RE_Q R^T - L^{-1}FR^T - RF^T L^{-T} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \tau \begin{pmatrix} w_1 \\ -w_2 \end{pmatrix}. \tag{15}$$

Note that the diagonal blocks are symmetric. Now multiply (15) by w^H thus obtaining

$$\begin{aligned} w_1^H E_Q w_1 + w_1^H(-E_Q R^T + F^T L^{-T})w_2 &= \tau \|w_1\|^2 \\ w_2^H(-E_Q R^T + F^T L^{-T})^T w_1 + w_2^H RE_Q R^T w_2 - 2\Re(w_2^H L^{-1}FR^T w_2) &= -\tau \|w_2\|^2 \end{aligned} \tag{16}$$

Subtracting the two equations yields two equations for the real and the imaginary part respectively (assuming $\|w\| = 1$).

$$\begin{aligned} w_1^H E_Q w_1 - w_2^H RE_Q R^T w_2 + 2\Re(w_2^H L^{-1}FR^T w_2) &= \tau_R \\ 2\Im(w_1^H(-E_Q R^T + F^T L^{-T})w_2) &= \tau_I \end{aligned} \tag{17}$$

If, instead, we add together the two equations in (16) we get for the imaginary part

$$0 = \tau_I \left(\|w_1\|^2 - \|w_2\|^2\right)$$

which gives for every complex eigenvalue $\|w_1\|^2 = \|w_2\|^2 = \dfrac{1}{2}$. Hence (17) provides a bound for the real and imaginary part of τ in terms of $e_Q = \|E_Q\|$ and e_J.

$$\begin{aligned} |\tau_R| &\leq |w_1^H E_Q w_1| + |w_2^H RE_Q R^T w_2| + 2|w_2^H L^{-1}FR^T w_2| \\ &\leq \|E_Q\| \left(\|w_1\|^2 + \|w_2\|^2\right) + 2\|w_2\|^2 \, \|L^{-1}FR^T\| = e_Q + e_J \\ |\tau_I| &\leq 2\|w_1\|\|w_2\|\|(-E_Q R^T + F^T L^{-T})\| \leq e_Q + e_J \end{aligned} \tag{18}$$

Our new proof has followed the ideas from the paper of Benzi and Simoncini [1]. The major difference is that in [1] the preconditioner uses the *exact* Jacobian A, while our analysis applies to the case when the *approximate* Jacobian $\tilde{A}$ is used.

In the special case of linear programming or separable quadratic programming, the Hessian of Lagrangian Q is a diagonal matrix, hence the preconditioner uses exact Hessian $D = Q$. The analysis simplifies in this case.

		nonseparable case		separable case	
case	eigenvector	Eig	bound	Eig	bound
1.	$u = 0, F^T y = 0$	R	$\lambda = 1$	R	$\lambda = 1$
1.	$u = 0, F^T y \neq 0$	R/C	$\|\lambda - 1\| \leq e_J$	R/C	$\|\lambda - 1\| \leq e_J$
2.	$u \neq 0, Bu \neq 0, Fu = 0$	R	$\lambda = 1$	R	$\lambda = 1$
2.	$u \neq 0, Bu \neq 0, Fu \neq 0$	R	$\lambda \in [\lambda_{\min}(K), \lambda_{\max}(K)]$	R	$\lambda = 1$
3.	$u \neq 0, Bu \neq 0, Fu \neq 0$	C	$\begin{cases} \|\Re(\lambda) - 1\| < e_Q + e_J \\ \|\Im(\lambda) - 1\| < e_Q + e_J \end{cases}$	C	$\|\lambda - 1\| \leq e_J$

Table 1: Types of eigenvalues in $\tilde{P}^{-1}H$. R stands for real and C for complex eigenvalues.

Corollary. Assume that $\tilde{A}$ has maximum rank. The eigenvalues of $\tilde{P}^{-1}H$ are either one or bounded by

$$|\lambda - 1| \leq e_J.$$

Proof. The eigenvalues of $\tilde{P}^{-1}H$ can be characterised in the same way as in Theorem 2 for the case $u = 0$ (they are either unit or bounded by $|\tau| < e_J$). If $u \neq 0$, the real eigenvalues must satisfy (12) with $K = I$, from which $\lambda = 1$; while for the complex ones, using $E_Q = 0$, the first equation of (16) simplifies to

$$|\tau| = \frac{|w_1^H F^T L^{-T} w_2|}{\|w_1\|^2} \leq \frac{\|w_1\| \, e_J \, \|w_2\|}{\|w_1\|^2} = e_J.$$

We summarise the classification of eigenvalues of preconditioned matrix in Table 1.

2.1 Bounds on complex eigenvalues

In this section we give further bounds on the modulus and real and imaginary part of the complex eigenvalues.

Theorem. If $x \neq 0$ the complex eigenvalues are bounded by

$$|\lambda - 1| \leq e_Q + \frac{1 + \sqrt{5}}{2} e_J \tag{19}$$

Proof. We refer again to system (9). Let us now decompose u into $u = u_0 + u_\perp$, where $\tilde{B}u_0 = 0$ and $u_0^T u_\perp = 0$, and set $p = u_\perp + \tilde{B}^T y$.

$$\|F^T y\| = \|F^T\| \frac{\|y\|}{\|\tilde{B}^T y\|} \|\tilde{B}^T y\| \leq \frac{\|F^T\|}{\sigma_1(\tilde{B})} (\|p\| + \|u_\perp\|)$$

Taking the norms, the first equation of (9) reads

$$\begin{aligned}
|\tau| &\leq \frac{\|(K-I)u\| + \|F^T y\|}{\|u_0\| + \|u_\perp + \tilde{B}^T y\|} && (20)\\
&\leq \frac{\|(K-I)u\|}{\|u_0\| + \|p\|} + \frac{\|F^T y\|}{\|u_0\| + \|p\|} && (21)\\
&\leq e_Q \frac{\|u_0\| + \|u_\perp\|}{\|u_0\| + \|p\|} + e_J \frac{\|p\| + \|u_\perp\|}{\|u_0\| + \|p\|} && (22)\\
&= e_J + \frac{(e_Q - e_J)\|u_0\| + (e_Q + e_J)\|u_\perp\|}{\|u_0\| + \|p\|} = b(\|p\|) && (23)
\end{aligned}$$

Function $b(\|p\|)$ is

$$\begin{cases}
\text{(a1)} & \text{decreasing} \quad \text{if } e_Q > e_J \\
\text{(b1)} & \text{decreasing} \quad \text{if } e_Q \leq e_J \text{ and } t = \dfrac{\|u_\perp\|}{\|u_0\|} > \dfrac{e_J - e_Q}{e_Q + e_J} \\
\text{(b2)} & \text{increasing} \quad \text{if } e_Q \leq e_J \text{ and } t \leq \dfrac{e_J - e_Q}{e_Q + e_J}
\end{cases}$$

so that if $e_Q > e_J$ then $b(\|p\|) \leq b(0) = e_Q(1+t) + e_J t \equiv f(t)$. If $e_Q \leq e_J$:

$$b(\|p\|) \leq \begin{cases}
b(0) = e_Q(1+t) + e_J t & \text{if} \quad t > \dfrac{e_J - e_Q}{e_Q + e_J} \\
b(\infty) = e_J & \text{if} \quad t \leq \dfrac{e_J - e_Q}{e_Q + e_J}
\end{cases} \equiv f(t)$$

From the second equation of (9), we obtain

$$|\tau| \leq \frac{\|F\| \|u_0 + u_\perp\|}{\|\tilde{B}u_\perp\|} \leq \frac{\|F\| \|u_0 + u_\perp\|}{\sigma_1(\tilde{B})\|u_\perp\|} \leq \frac{\|F\|}{\sigma_1(\tilde{B})} \left(1 + \frac{\|u_0\|}{\|u_\perp\|}\right) = e_J \left(1 + \frac{1}{t}\right) \equiv g(t).$$

Since τ must satisfy $|\tau| \leq f(t)$ and $|\tau| \leq g(t)$ for every t, and given that $f(t)$ is nondecreasing while $g(t)$ is decreasing, also $|\tau| \leq f(\bar{t})$ where $\bar{t}$ is s.t. $f(\bar{t}) = g(\bar{t})$. The intersection between f and g in the case $e_Q \leq e_J$ can take place only if $t > \dfrac{e_J - e_Q}{e_Q + e_J}$ (case b1) since $e_j < g(t)$. Therefore $f(t) = e_Q(1+t) + e_J t$.

Now, setting $z = \dfrac{e_J}{e_Q}$, we have that

$$f(t) = g(t) \iff 1 + t + zt = z + \frac{z}{t} \iff t^2(1+z) + (1-z)t - z = 0$$

then

$$\bar{t} = \frac{z - 1 + \sqrt{5z^2 + 2z + 1}}{2(1+z)}$$

Now

$$|\tau| < f(\bar{t}) = e_Q + (1+z)\bar{t}e_Q = e_Q + \frac{1+z}{z}\bar{t}e_J \quad (24)$$

$$= e_Q + \frac{z - 1 + \sqrt{5z^2 + 2z + 1}}{2z} e_J \leq e_Q + \frac{1+\sqrt{5}}{2} e_J. \quad (25)$$

The last inequality follows observing that

$$h(z) = \frac{z - 1 + \sqrt{5z^2 + 2z + 1}}{2z} = \frac{1}{2}\left(1 - \frac{1}{z} + \sqrt{5 + \frac{2}{z} + \frac{1}{z^2}}\right)$$

is an increasing function in $(0, +\infty)$ bounded by $\frac{\sqrt{5}+1}{2}$.

3 Numerical results

To effectively compute the preconditioner $\tilde{P}$ we reduce the augmented system to the normal equations $\tilde{A}D^{-1}\tilde{A}^T$, compute the Cholesky factorization

$$\tilde{A}D^{-1}\tilde{A}^T = L_0 D_0 L_0^T,$$

and use:

$$\tilde{P} = \begin{bmatrix} D & \tilde{A}^T \\ \tilde{A} & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ \tilde{A}D^{-1} & I \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & -\tilde{A}D^{-1}\tilde{A}^T \end{bmatrix} \begin{bmatrix} I & D^{-1}\tilde{A}^T \\ 0 & I \end{bmatrix}$$

$$= \begin{bmatrix} I & 0 \\ \tilde{A}D^{-1} & L_0 \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & -D_0 \end{bmatrix} \begin{bmatrix} I & D^{-1}\tilde{A}^T \\ 0 & L_0^T \end{bmatrix}. \quad (26)$$

We have employed the iterative method QMRs [5], which is particularly well-suited for symmetric indefinite systems. We have compared four alternative methods to solve (1): direct approach as implemented in HOPDM [6], two variants of preconditioned conjugate gradients using exact constraint preconditioners of form (2) as developed in [3], and the new preconditioner $\tilde{P}$ by (26) in which approximate Jacobian $\tilde{A}$ is used.

We have solved several problems from public domain collections of quadratic programs. To avoid reporting excessive numerical results, we have selected a subset of 7 representative quadratic programs for which we give detailed solution statistics. Problems `sqp2500_*` have been made available to us by Professor Hans Mittelmann. Problems `AUG3D*` originate from CUTE library and can be retrieved for example from `http://www.sztaki.hu/~meszaros/public_ftp/qpdata/cute/`.

3.1 Results with constant $\tilde{A}$

We start the analysis from the statistics of problems used in our computations. In Table 2 we report problem sizes m, n, the number of nonzero elements in matrix A, the number of off-diagonal nonzero elements in Q and in the factorization of the complete matrix H, nnz(L).

problem	m	n	nnz(A)	nnz(Q)	nnz(L)
AUG3D	15625	68053	112981	0	2446430
AUG3DQP	15625	68053	85169	0	1507295
AUG3DC	27000	103107	181320	0	5269968
AUG3DCQP	27000	103107	181286	0	5336157
sqp2500_1	2000	4000	52321	738051	3124093
sqp2500_2	2000	4000	52319	14345	3504910
sqp2500_3	4500	7000	115073	738051	3219994

Table 2: Values of m, n, nonzeros in A, off-diagonal nonzeros in Q and in the triangular factors L for augmented matrix nnz(L).

The iterative methods QMRs and PCG have been stopped using a tolerance tol on the relative residual $\frac{\|r_k\|}{\|b\|} \leq \texttt{tol} = 10^{-2}, 10^{-4}$ and a limit of iterations itmx $\in [50, 100]$.

All tests have been run on an Intel Xeon PC 2.80 GHz with 2 GB RAM. We have used the pure FORTRAN version of the solver and we have compiled it with the g77 compiler with -O4 option. The CPU times are measured in seconds.

In the definition of preconditioner $\tilde{P}$ we used the following dropping rule to determine matrix E:

$$e_{ij} = \begin{cases} a_{ij} & \text{if } |a_{ij}| < \texttt{drop}\ \|A_j\| \ \text{ AND } \ |i-j| > \texttt{nb} \\ 0 & \text{otherwise} \end{cases}$$

where with A_j we denote the jth column of A.
In other words, we drop an element from matrix A if it is below a prescribed tolerance and outside a fixed band. The first requirement prevents $\|E\|$ from becoming too large with consequent going away of the eigenvalues from the unity (see the bound (10) in Theorem 1). The second requirement attempts to control the fill-in of AA^T and hence of its Cholesky factor L.

Table 3 collects the results of HOPDM runs on all separable test examples for the direct approach (factorization of H) and the QMRs (and PCG) iterative methods with preconditioner $\tilde{P}$ with the optimal combination of parameters, namely itmx , tol , nb , drop , experimentally found after extensive testing.

Table 4 provides the same outcome on the sqp2500_⋆ problems. For these tests, we also report the performance of the PCG preconditioned P (“exact” preconditioner,

Problem AUG3D									
solver	itmx	tol	nb	drop	CPU	nnz(E)	nnz(L)	Its	LinIt
direct					62.58		2446430	13	
QMRs ($\tilde{P}$)	100	1.e-2	100	1.0	11.49	89483	184	12	303
PCG ($\tilde{P}$)	100	1.e-2	100	1.0	16.83	89483	184	13	894
Problem AUG3DQP									
solver	itmx	tol	nb	drop	CPU	nnz(E)	nnz(L)	Its	LinIt
direct					12.14		1507295	12	
QMRs ($\tilde{P}$)	100	1.e-2	10	1.0	3.25	69828	0	13	53
PCG ($\tilde{P}$)	100	1.e-2	10	1.0	2.65	69828	0	13	80
Problem AUG3DC									
solver	itmx	tol	nb	drop	CPU	nnz(E)	nnz(L)	Its	LinIt
direct					240.32		5269968	17	
QMRs ($\tilde{P}$)	50	1.e-2	10	1.0	248.63	54966	2522599	34	519
PCG ($\tilde{P}$)	50	1.e-2	10	1.0	299.53	54966	2522599	29	2478
Problem AUG3DCQP									
solver	itmx	tol	nb	drop	CPU	nnz(E)	nnz(L)	Its	LinIt
direct					339.28		5336157	24	
QMRs ($\tilde{P}$)	50	1.e-2	10	1.0	288.18	54966	2563585	39	590
PCG ($\tilde{P}$)	50	1.e-2	10	1.0	265.23	54966	2563585	25	2263

Table 3: Performance of the proposed preconditioner with optimal combination of the parameters vs direct solver. Separable problems AUG3D*.

obtained by using $\tilde{A} = A$ in (3) and (26), respectively). We report in Tables 3 and 4 the total CPU time, the fill-in of the Cholesky factor of the preconditioner, the number of interior point iterations, Its, and the overall number of iterations in the iterative solver, LinIt.

As for the separable problems, the iterative approach produces mixed results. For some of them there is a significant reduction of the CPU time. On problems AUG3DC and AUG3DCQP, however, dropping some elements from matrix A does not produce a sufficient reduction of the fill-in of L. Hence, the cost of a single QMRs/PCG iteration is comparable to that of the direct solution of the linear system.

Regarding the sqp2500_* problems, the situation is different. The QMRs preconditioned with "best" $\tilde{P}$ outperforms direct solver as well as the PCG method.

The different behaviour of the proposed approach on the two classes of problems may be put in connection with the nonzero pattern and values of the matrices involved. In the AUG$*$ problems, matrix A has a banded nonzero structure, moreover most of its elements have the same absolute value. For such problems, our dropping strategy tends to remove either a very small or a very large number of nonzero elements. In the former case the preconditioner does not provide sufficient acceleration of the iterative method, whilst in the latter the high density of the preconditioner factors weighs down

Problem `sqp2500_1`									
solver	`itmx`	`tol`	`nb`	`drop`	CPU	nnz(E)	nnz(L)	Its	LinIt
direct					167.52		3124093	15	
PCG (P)	20	1.e-2			132.96		1909672	19	539
QMRs ($\tilde{P}$)	50	1.e-2	10	1.0	32.59	49390	41	18	1481
PCG ($\tilde{P}$)	50	1.e-2	100	1.0	38.43	51320	11303	18	2030

Problem `sqp2500_2`									
solver	`itmx`	`tol`	`nb`	`drop`	CPU	nnz(E)	nnz(L)	Its	LinIt
direct					206.32		3504910	16	
PCG (P)	20	1.e-2			120.37		1909275	19	499
QMRs ($\tilde{P}$)	50	1.e-2	10	1.0	6.11	49413	37	20	1740
PCG ($\tilde{P}$)	50	1.e-2	100	0.5	5.45	51319	16125	18	2039

Problem `sqp2500_3`									
solver	`itmx`	`tol`	`nb`	`drop`	CPU	nnz(E)	nnz(L)	Its	LinIt
direct					215.93		3216994	19	
PCG (P)	20	1.e-2			1447.23		9874267	24	308
QMRs ($\tilde{P}$)	75	1.e-4	10	1.0	82.37	112034	50	24	3646
PCG ($\tilde{P}$)	50	1.e-2	100	1.0	68.72	107576	14153	31	3250

Table 4: Performance of the proposed preconditioner with optimal combination of the parameters vs PCG preconditioned with P_1 and P_2 and the direct solver. Non-separable quadratic problems `sqp2500_*`.

the cost of an iteration of the Krylov subspace method.

We also note that the PCG method gives convergence in all the test cases even with the inexact preconditioner. In some cases (problems `AUG3DQP, AUG3DCQP, sqp2500_2, sqp2500_3`) the PCG with preconditioner $\tilde{P}_2$ also leads to the best results in terms of total CPU time.

4 Selective computing of $\tilde{A}$

In this section we present some preliminary results obtained using a dinamyc dropping in the Jacobian matrix A.

The idea is based observing that, at convergence, one of the following conditions occurs

1. (primal variables) if $x_j \in \mathcal{B}$ then we have $x_j \to \hat{x}_j > 0$ and $s_j \to 0$ and so $\Theta_j \to \infty$, i.e. $\Theta_j^{-1} \to 0$;

2. (non primal variables) if $x_j \in \mathcal{N}$ then we have $x_j \to 0$ and $s_j \to \hat{s}_j > 0$ and so $\Theta_j \to 0$, i.e. $\Theta_j^{-1} \to \infty$;

where $\Theta_j = \frac{x_j}{s_j}$, $\mathcal{B}$ denotes the set of the primal solution variables and $\mathcal{N}$ denotes the non primal variables.

This means that at the convergence if the $\Theta_j^{-1} > \Theta_{\tt drop}$ (in the test we use $\Theta_{\tt drop} = 10^{-4}$) the j–column of the matrix A is less significant and can be removed from the computation. Moreover, in order to avoid the computation of the preconditioner at each iteration we have introduced a safety factor on the percent number of removed columns (here we use 5%). In detail, we rebuild the preconditioner only if the current number of removed columns, computed using the previous strategy, differs from a percent value of the total number of columns.

As an example we consider the problem `25FV47`. We report in Table 5 the same parameters as in Table 2. The comparison with and without dinamic preconditioner is reported in Table 6. These preliminary results are encouraging.

problem	m	n	nnz(A)	nnz(Q)	nnz(L)
`25FV47`	820	1571	10464	59053	20296

Table 5: Values of m, n, nonzeros in A, off-diagonal nonzeros in Q and in the triangular factors L for augmented matrix nnz(L).

Without dinamic rebuilding									
solver	`itmx`	`tol`	`nb`	`drop`	CPU	nnz(E)	nnz(L)	Its	LinIt
QMRs ($\tilde{P}$)	100	1.e-2	700	0.05	6.35	28	20317	50	471
With dinamic rebuilding (no. of total rebuild = 16) Data refer to the last preconditioner									
solver	`itmx`	`tol`	`nb`	`drop`	CPU	nnz(E)	nnz(L)	Its	LinIt
QMRs ($\tilde{P}$)	100	1.e-2	700	0.05	5.8	6800	11341	54	408

Table 6: Performance of the proposed preconditioner for the problem `25FV47`.

5 Conclusions

We have provided in this paper the analysis of inexact constraint preconditioner for equality constrained optimization problems. Dropping some of the elements in the Jacobian matrix A produces a significant reduction of the fill-in of the Cholesky factor of the preconditioner thus speeding-up the cost of a single iteration of the Krylov subspace method of choice. The spectral analysis of the preconditioned matrix reveals that a large number of eigenvalues are one or positive and bounded by those of $D^{-1}Q$. The distance of the remaining eigenvalues form unity is proven to be bounded in terms of the norm of the dropping matrix E. Some numerical results onto a number of large quadratic problems demonstrate that the new approach is an attractive alternative for direct approach and for exact constraint preconditioners.

References

[1] M. Benzi and V. Simoncini, *On the eigenvalues of a class of saddle point matrices*, Numer. Math., 103 (2006), pp. 173–196.

[2] L. Bergamaschi, J. Gondzio, M. Venturin, and G. Zilli, *Inexact constraint preconditioners for linear systems arising in interior point methods*, Comput. Optim. Appl., 36 (2007), pp. 136–147.

[3] L. Bergamaschi, J. Gondzio, and G. Zilli, *Preconditioning indefinite systems in interior point methods for optimization*, Computational Optimization and Applications, 28 (2004), pp. 149–171.

[4] L. Bergamaschi, M. Venturin, and G. Zilli, *Inexact constraint preconditioners for optimization problems*, in Proceedings of the Fifth Int. Conf. on Engineering Comput. Technology. Paper # 89, B. H. V. Topping, G. Montero, and R. Montenegro, eds., Civil-Comp Press, Stirlingshire, UK, (2006), pp. 1–10. doi:10.4203/ccp.84.89

[5] R. W. Freund and N. M. Nachtigal, *Software for simplified Lanczos and QMR algorithms*, Applied Numerical Mathematics, 19 (1995), pp. 319–341.

[6] J. Gondzio, *HOPDM (version 2.12) – a fast LP solver based on a primal-dual interior point method*, European Journal of Operational Research, 85 (1995), pp. 221–225.

[7] N. I. M. Gould, M. E. Hribar, and J. Nocedal, *On the solution of equality constrained quadratic problems arising in optimization*, SIAM Journal on Scientific Computing, 23 (2001), pp. 1375–1394.

[8] C. Keller, N. I. M. Gould, and A. J. Wathen, *Constraint preconditioning for indefinite linear systems*, SIAM Journal on Matrix Analysis and Applications, 21 (2000), pp. 1300–1317.

[9] L. Lukšan and J. Vlček, *Indefinitely preconditioned inexact Newton method for large sparse equality constrained nonlinear programming problems*, Numerical Linear Algebra with Applications, 5 (1998), pp. 219–247.