Inexact Constraint Preconditioners for Linear Systems Arising in Interior Point Methods[1]

Luca Bergamaschi [2] [3] Jacek Gondzio [4] Manolo Venturin [5] Giovanni Zilli [6]

Technical Report MS-05-002, November 4, 2005, revised March 1, 2006, and May 22, 2006

Abstract. Issues of indefinite preconditioning of reduced Newton systems arising in optimization with interior point methods are addressed in this paper. Constraint preconditioners have shown much promise in this context. However, there are situations in which an unfavorable sparsity pattern of Jacobian matrix may adversely affect the preconditioner and make its inverse representation unacceptably dense hence too expensive to be used in practice. A remedy to such situations is proposed in this paper. An approximate constraint preconditioner is considered in which sparse approximation of the Jacobian is used instead of the complete matrix. 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. Interior point methods for linear, quadratic or nonlinear programming are the key optimization methodology. Their theory [12] and implementation [1] is well understood. When quadratic or nonlinear programming problems subject to equality constraints are tackled each iteration of an interior point method requires the following linear system of equations to be solved

$$Hx = b \qquad \text{where} \qquad H = \begin{bmatrix} Q & A^T \\ A & \end{bmatrix}, \tag{1.1}$$

where $Q \in \mathcal{R}^{n\times n}$ is the Hessian of Lagrangian and $A \in \mathcal{R}^{m\times n}$ is the Jacobian of constraints. The matrix Q arising in interior point applications has form $Q = Q_0 + \Theta$, where diagonal scaling matrix $\Theta \in \mathcal{R}^{n\times n}$ reflects the presence of barrier terms for primal variables. We assume that Q_0 and in consequence Q is positive definite and A has full row rank.

Linear systems of this type appear commonly in partial differential equations where they are known as the saddle point problems. A vast amount of literature exists which address the solution of such systems in the context of differential equations. We draw readers attention to recent survey [2] which provides a comprehensive discussion of these issues and guides through the literature on the subject. A common practice in differential equations community is to solve saddle point problems with *iterative* methods.

There has been growing interest in recent years in the use of iterative methods to solve system (1.1) arising in optimization context. This is because certain large instances of (1.1) 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 [4, 9, 10, 11] to mention a few. They have a common feature of constructing the approximation to (1.1) by simplifying its upper left block, namely by applying the preconditioner of the form

$$P = \begin{bmatrix} D & A^T \\ A & \end{bmatrix}. \tag{1.2}$$

Usual choices for matrix D are to keep it block-diagonal or diagonal. The latter situation has been studied for example in [4]: 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.

[1] accepted for publication in **Computational Optimization and Applications.**

[2] Corresponding author

[3] Department of Mathematical Methods and Models for Scientific Applications, University of Padua, Italy, berga@dmsa.unipd.it

[4] School of Mathematics, University of Edinburgh, Scotland, J.Gondzio@ed.ac.uk

[5] Department of Pure and Applied Mathematics, University of Padua, Italy, mventuri@math.unipd.it

[6] Department of Mathematical Methods and Models for Scientific Applications, University of Padua, Italy, zilli@dmsa.unipd.it

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 this short note we look for a further simplification of the preconditioner: we replace Jacobian matrix A with a (sparse) approximation $\tilde{A}$. Our approach goes along the direction of the work of Coleman and Verma [5] who showed how to use the reduced preconditioned conjugate gradient method in this context. In a recent paper Benzi and Simoncini [3] analyzed the class of saddle point preconditioners which avoid constructing matrix AA^T and use its approximation instead. In another recent paper Dollar and Wathen [6] used preconditioners based on incomplete Schilders' factorization.
We propose to use the preconditioner of the following form

$$\tilde{P} = \begin{bmatrix} D & \tilde{A}^T \\ \tilde{A} & \end{bmatrix}, \tag{1.3}$$

where D is an approximation of Q and $\tilde{A}$ is an approximation of A. We extend the analysis of [4] to this case and provide a complete characterisation of the spectrum of $\tilde{P}^{-1}H$. Our findings can be summarised as follows. Suppose both A and $\tilde{A}$ have full rank m and the error of approximation $E = A - \tilde{A}$ has rank p, where $0 \leq p \leq m$. The use of approximation of A allows for a presence of complex eigenvalues in the preconditioned matrix. There are at most $2p$ complex eigenvalues and they are bounded. We derive a bound which depends on the error of approximation E and the stability properties of approximation matrix $\tilde{A}$. The bound is constructive and its interpretation helps to avoid numerical difficulties in the implementation of the method. We have implemented the approach proposed in this note in HOPDM interior point solver [8] and we have tested it on a class of quadratic programming problems.
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. Finally, in Section 4 we give our conclusions.

2. Spectral Analysis. We consider preconditioner (1.3) which approximates Hessian matrix Q with matrix D and Jacobian matrix A with (sparser) matrix $\tilde{A}$. To guarantee nonsingularity of the preconditioner we assume that D is invertible and $\tilde{A}$ has full row rank that is $rank(\tilde{A}) = m \leq n$. In the analysis we use the error of Jacobian approximation $E = A - \tilde{A}$, and denote as $\tilde{\sigma}_1$ the smallest singular value of $\tilde{A}$.
The following theorem characterises the eigenvalues of $\tilde{P}^{-1}H$ in the case $n > m$. This case is common for optimization applications.

THEOREM 2.1. *Assume A and $\tilde{A}$ have maximum rank. The eigenvalues of $\tilde{P}^{-1}H$ are either one (with multiplicity at least $2m - 2p$) or real positive and bounded by*

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

or bounded by

$$|\lambda - 1| \leq \frac{\|E\|}{\tilde{\sigma}_1}. \tag{2.1}$$

Proof. The eigenvalues and eigenvectors of $\tilde{P}^{-1}H$ must satisfy

$$\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} D & (A-E)^T \\ A-E & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.2}$$

or

$$\begin{cases} Qx + A^T y &= \lambda D x + \lambda A^T y - \lambda E^T y \\ Ax &= \lambda A x - \lambda E x \end{cases}. \tag{2.3}$$

We now analyze a number of cases depending on x and y.

1. $\boxed{x = 0}$

(a) $\boxed{E^T y = 0}$ There are $m - p$ linearly independent vectors y such that $E^T y = 0$. The first equation simplifies to

$$A^T y = \lambda A^T y$$

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

(b) $\boxed{E^T y \neq 0}$ The first equation reads

$$A^T y = \lambda A^T y - \lambda E^T y\,.$$

Writing $\lambda = 1 + \epsilon$ we can bound ϵ in terms of $\|E\|$ as

$$|\epsilon| = \frac{\|E^T y\|}{\|\tilde{A}^T y\|} \leq \|E\| \frac{\|y\|}{\|\tilde{A}^T y\|} \leq \frac{\|E\|}{\tilde{\sigma}_1}\,. \tag{2.4}$$

2. $\boxed{x \neq 0}$

(a) $\boxed{Ax = 0}$ There are $n - m$ linearly independent vectors satisfying $Ax = 0$. From the second equation we have $\lambda E x = 0$, hence premultiplying the first equation by x^T we obtain

$$x^T Q x = \lambda x^T D x\,.$$

From the positive definitness of D and Q it follows that this eigenvalue is bounded by the extremal (positive) eigenvalues of $D^{-1}Q$: $\lambda_{\min}(D^{-1}Q) \leq \lambda \leq \lambda_{\max}(D^{-1}Q)$.

(b) $\boxed{Ax \neq 0, Ex = 0}$ There are $n - p - (n - m) = m - p$ linearly independent vectors satisfying $Ex = 0$. Now, the second equation reads

$$Ax = \lambda A x$$

giving again unit eigenvalues.

(c) $\boxed{Ax \neq 0, Ex \neq 0}$ Let us multiply the first equation by x^H and the second by y^H. We obtain the following system

$$\left\{ \begin{array}{lcl} x^H Q x + x^H A^T y & = & \lambda x^H D x + \lambda x^H \tilde{A}^T y \\ y^H A x & = & \lambda y^H \tilde{A} x \end{array} \right. . \tag{2.5}$$

Now, setting $c_1 = x^H Q x, c_2 = x^H D x; c_1, c_2 \in \mathbb{R}$ and $d_1 = x^H A^T y, d_2 = x^H \tilde{A}^T y$ and adding the two equations we obtain

$$c_1 + 2\Re(d_1) = \lambda c_2 + 2\lambda \Re(d_2)$$

If $c_2 + 2\Re(d_2) = 0$ then λ can be complex and from the second equation we again get the bound (2.4):

$$|\epsilon| = \frac{\|Ex\|}{\|\tilde{A}x\|} \leq \|E\| \frac{\|x\|}{\|\tilde{A}x\|} \leq \frac{\|E\|}{\tilde{\sigma}_1}\,. \tag{2.6}$$

If $c_2 + 2\Re(d_2) \neq 0$ then $\lambda \in \mathbb{R}$. Hence, subtracting the transpose of the second equation from the first one, and using $\lambda = \bar{\lambda}$, $(x, y)^H = (x, y)^T$, we obtain the same bound as in 2(a).

$$x^T Q x = \lambda x^T D x.$$

□

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.

COROLLARY 2.2. *Assume that $\tilde{A}$ has maximum rank. The eigenvalues of $\tilde{P}^{-1}H$ are either one (with multiplicity at least $n+m-2p$) or bounded by*

$$|\lambda - 1| \leq \frac{\|E\|}{\tilde{\sigma}_1}.$$

Proof. The eigenvalues and eigenvectors of $\tilde{P}^{-1}H$ must satisfy

$$\begin{bmatrix} D & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} D & (A-E)^T \\ A-E & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.7}$$

or

$$\begin{cases} Dx + A^T y &= \lambda Dx + \lambda A^T y - \lambda E^T y \\ Ax &= \lambda Ax - \lambda Ex \end{cases} \tag{2.8}$$

The eigenvalues of $\tilde{P}^{-1}H$ can be characterised in the same way as in Theorem 2.1 except for cases 2(a) and 2(c) with $x^H Dx + 2\Re(x^H \tilde{A}^T y) \neq 0$. In both these cases the eigenvalue λ must satisfy

$$x^T Dx = \lambda x^T Dx, \qquad \text{with } x \neq 0$$

from which $\lambda = 1$. Then the unit eigenvalue has multiplicity $n - m + i$, $0 \leq i \leq p$, more than in the general (nonseparable) QP case. □

We summarise the classification of eigenvalues of preconditioned matrix in table below.

TABLE 2.1
Types of eigenvalues in $\tilde{P}^{-1}H$.

		nonseparable case		separable case	
Case	eigenvector	eigenvalue	multiplicity	eigenvalue	multiplicity
1(a)	$x = 0, E^T y = 0$	unit	$m-p$	unit	$m-p$
1(b)	$x = 0, E^T y \neq 0$	complex	p	complex	p
2(a)	$x \neq 0, Ax = 0$	real positive	$n-m$	unit	$n-m$
2(b)	$x \neq 0, Ax \neq 0, Ex = 0$	unit	$m-p$	unit	$m-p$
2(c)	$x \neq 0, Ax \neq 0, Ex \neq 0$	complex	$p-i$	complex	$p-i$
		real positive	i	unit	i

The analysis in Theorem 2.1 suggests that a good preconditioner should (i) keep $rank(E) = p$ as small as possible, and (ii) keep $\tilde{\sigma}_1$ as large as possible. The first requirement helps to limit the number of complex eigenvalues and the second helps to keep the bound on $|\lambda - 1|$ small.
Following the arguments in [4], after dropping all off-diagonal elements from Q and using sparser $\tilde{A}$ we expect an important gain in the sparsity of the Cholesky-like factor of preconditioner (1.3). Moreover, as in [4], we can exploit the diagonal form of D when computing the inverse representation of $\tilde{P}$. Indeed, we open the possibility of reducing the augmented system to the normal equations form whenever the latter offers any advantages. Summing up, we can choose one of the following forms.
Preconditioner 1 (AS):

$$\tilde{P}_1 = \begin{bmatrix} D & \tilde{A}^T \\ \tilde{A} & 0 \end{bmatrix} = L \Lambda L^T. \tag{2.9}$$

Preconditioner 2 (NE): Reduce the system to 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:

$$\tilde{P}_2 = \begin{bmatrix} D & \tilde{A}^T \\ \tilde{A} & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ \tilde{A}D^{-1} & I \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & -\tilde{A}D^{-1}\tilde{A}^T \end{bmatrix} \begin{bmatrix} I & D^{-1}\tilde{A}^T \\ 0 & I \end{bmatrix}$$

$$= \begin{bmatrix} I & 0 \\ \tilde{A}D^{-1} & L_0 \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & -D_0 \end{bmatrix} \begin{bmatrix} I & D^{-1}\tilde{A}^T \\ 0 & L_0^T \end{bmatrix}. \tag{2.10}$$

The two forms differ only in the implementation, being mathematically equivalent. The spectral analysis of the preconditioned matrix presented earlier applies to both. Computational results presented below use the second form of the preconditioner.

3. Numerical results. We have employed the iterative method QMRs [7], which is particularly well-suited for symmetric indefinite systems. We have compared four alternative methods to solve (1.1): direct approach as implemented in HOPDM [8], two variants of preconditioned conjugate gradients using exact constraint preconditioners of form (1.2) as developed in [4] (P_1 decomposes indefinite augmented system while P_2 uses its reduced normal equations form), and the new preconditioner $\tilde{P}_2$ by (2.10) 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/`.

We start the analysis from the statistics of problems used in our computations. In Table 3.1 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).

TABLE 3.1
Values of m, n, nonzeros in A, off-diagonal nonzeros in Q and in the triangular factors L for augmented matrix 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

The iterative method QMRs has 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 $\texttt{itmax} \in [50, 100]$.

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:

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

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 (2.4) in Theorem 2.1). The second requirement attempts to control the fill-in of AA^T and hence of its Cholesky factor L.

Table 3.2 collects the results of HOPDM runs on all separable test examples for the direct approach (factorization of H) and the QMRs method with preconditioner $\tilde{P}$ with the optimal combination of parameters, namely `itmax, tol, nband, drop`, experimentally found after extensive testing.

TABLE 3.2
Performance of the proposed preconditioner with optimal combination of the parameters vs direct solver. Separable problems `AUG3D*`.

Problem	solver	itmax	tol	nband	drop	CPU	nnz(E)	nnz(L)	Its	LinIt
AUG3D	direct					62.58		2446430	13	
	QMRs ($\tilde{P}_2$)	100	1.e-2	100	1.0	11.49	89483	184	12	303
AUG3DQP	direct					12.14		1507295	12	
	QMRs ($\tilde{P}_2$)	100	1.e-2	10	1.0	3.25	69828	0	13	53
AUG3DC	direct					240.32		5269968	17	
	QMRs ($\tilde{P}_2$)	50	1.e-2	10	1.0	248.63	54966	2522599	34	519
AUG3DCQP	direct					339.28		5336157	24	
	QMRs ($\tilde{P}_2$)	50	1.e-2	10	1.0	288.18	54966	2563585	39	590

TABLE 3.3
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_*`.

Problem	solver	itmax	tol	nband	drop	CPU	nnz(E)	nnz(L)	Its	LinIt
sqp2500_1	direct					167.52		3124093	15	
	PCG (P_1)	20	1.e-2			141.03		2029613	19	539
	PCG (P_2)	20	1.e-2			132.96		1909672	19	539
	QMRs ($\tilde{P}_2$)	50	1.e-2	10	1.0	32.59	49390	41	18	1481
sqp2500_2	direct					206.32		3504910	16	
	PCG (P_1)	20	1.e-2			130.66		2055283	19	499
	PCG (P_2)	20	1.e-2			120.37		1909275	19	499
	QMRs ($\tilde{P}_2$)	50	1.e-2	10	1.0	6.11	49413	37	20	1740
sqp2500_3	direct					215.93		3216994	19	
	PCG (P_1)	20	1.e-2			278.22		3186754	23	291
	PCG (P_2)	20	1.e-2			1447.23		9874267	24	308
	QMRs ($\tilde{P}_2$)	75	1.e-4	10	1.0	82.37	112034	50	24	3646

Table 3.3 provides the same outcome on the `sqp2500_*` problems. For these tests, we also report the performance of the PCG preconditioned with both P_1 and P_2 ("exact" preconditioners, obtained by using $\tilde{A} = A$ in (2.9) and (2.10), resp.). We report in Tables 3.2 and 3.3 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 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 the cost of a QMRs iteration.

3.1. Detailed analysis on results of problem `sqp2500_2`. To show how the `drop` and `nband` values may affect the distribution of the eigenvalues of $\tilde{P}^{-1}H$, in Table 3.4 we first report, for some combination of the parameters, a few characteristics of the preconditioned matrix, namely the number of unit eigenvalues, the maximum distance from the unity ($|\epsilon|$, see the proof of Theorem 2.1), and also the smallest real part of all the eigenvalues. In correspondence to the same cases of Table 3.4 we have plotted all the eigenvalues of

TABLE 3.4
Spectral properties of the preconditioned matrices for problem `sqp2500_2`.

nband	drop	p	nnz(E)	ones	$\lvert\epsilon\rvert$	min$\{\Re(\lambda)\}$
∞	0	0	0	4007	0.33	0.79
100	0.01	909	1042	2005	0.33	0.79
10	0.10	1996	12534	1551	1.64	0.23
100	0.25	2000	30157	1550	5.08	0.04

the preconditioned matrix on the complex plane. In the "no-drop" case (exact constraint preconditioner), the eigenvalues are all real (the ones are $4007 > 2m = 4000$, as expected). With $E \neq 0$, the number of unit eigenvalues is smaller but still remains important. Increasing the `drop` parameter, also $|\epsilon|$ increases, but the real part of eigenvalues still remains bounded away from zero.
To further analyze the influence of the parameters on the performance of the preconditioner, we report in Table 3.5 the results of a number of runs on problem `sqp2500_2` with different values of `itmax`, `nband` and `drop`. The value of `tol` has been kept constant to 10^{-2}. From the table, we can observe that too large `nband` values or too small `drop` values results in a large number of nonzero elements in the Cholesky factor. On the other hand, fast convergence of the iterative method is achieved even with an almost zero fill-in of the triangular factor L.
The maximum number of iterations `itmax` of iterative solver should be carefully selected. In our test problems we found that `itmax` = 100 was sufficient to achieve convergence for every choice of the preconditioner $\tilde{P}$. However, in many cases a smaller value of `itmax` produced cheaper solution of linear system (1.1) without affecting overall convergence of the interior point method.

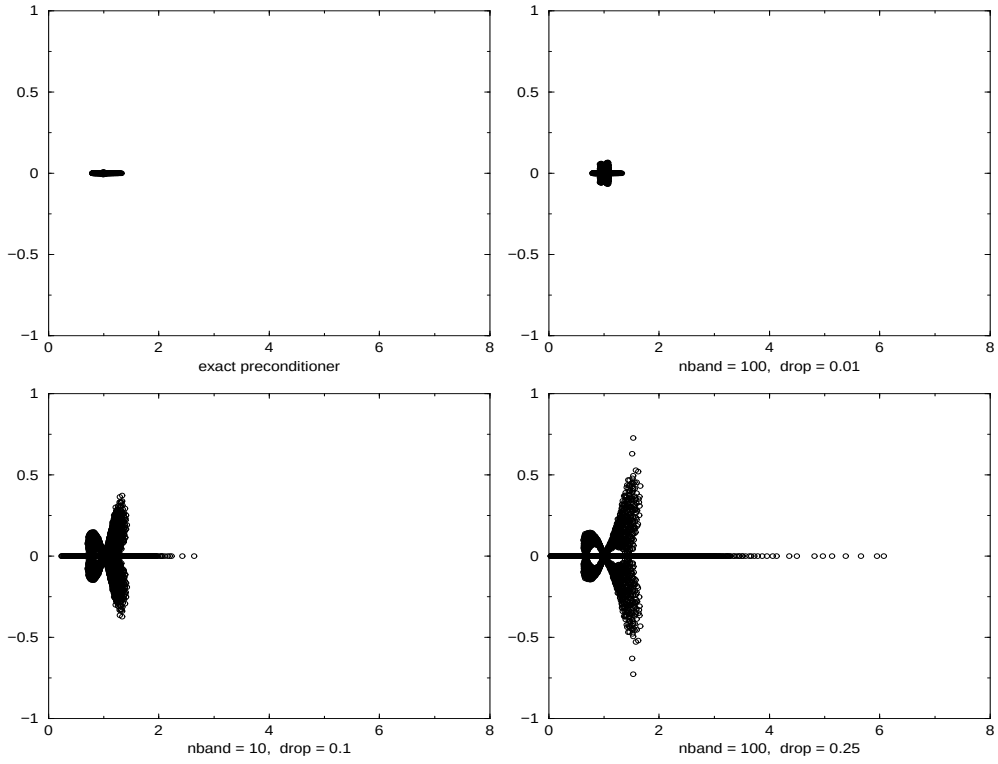


FIG. 3.1. *Distribution of the eigenvalues of* $\tilde{P}^{-1}H$ *in the complex plane for problem* `sqp2500_2` *with different combinations of the parameters* `nband`, `drop`.

4. Conclusions. We have provided in this paper the analysis of inexact constraint preconditioner for equality constrained optimization problems. We have reported preliminary numerical results which demonstrate that the new approach is an attractive alternative for direct approach and for exact constraint

TABLE 3.5
Performance of preconditioner $\tilde{P}_2$ with some combinations of the parameters. Problem `sqp2500_2`.

itmax	tol	nband	drop	CPU	nnz(E)	nnz(L)	Its	LinIt
50	1.e-2	10	1.0	6.11	49413	37	20	1740
75	1.e-2	10	1.0	7.93	49413	37	19	2323
100	1.e-2	10	1.0	9.37	49413	37	19	2769
50	1.e-2	10	0.5	6.52	49319	69	21	1847
75	1.e-2	10	0.5	7.98	49319	69	19	2322
100	1.e-2	10	0.5	9.40	49319	69	19	2775
50	1.e-2	10	0.25	101.52	32434	1289254	20	1747
100	1.e-2	10	0.25	116.76	32434	1289254	19	2468
50	1.e-2	10	0.1	132.38	12534	1821164	19	1148
75	1.e-2	10	0.1	132.52	12534	1821164	19	1148
100	1.e-2	10	0.1	132.56	12534	1821164	19	1148
50	1.e-2	100	1.0	6.94	45972	12888	20	1746
75	1.e-2	100	1.0	7.24	45972	12888	16	1861
100	1.e-2	100	1.0	8.57	45972	12888	16	2212
50	1.e-2	100	0.5	7.34	45884	19829	20	1744
75	1.e-2	100	0.5	7.67	45884	19829	16	1861
100	1.e-2	100	0.5	9.01	45884	19829	16	2220
50	1.e-2	100	0.25	110.64	30157	1405559	20	1670
100	1.e-2	100	0.25	103.49	30157	1405559	16	1888
50	1.e-2	100	0.1	133.40	11647	1832392	19	1164
75	1.e-2	100	0.1	134.33	11647	1832392	19	1166
100	1.e-2	100	0.1	133.56	11647	1832392	19	1166
50	1.e-2	100	0.01	122.44	1186	1901598	19	583
50	1.e-2	1000	1.0	139.96	16250	1763114	19	1606
75	1.e-2	1000	1.0	156.28	16250	1763114	19	2148
100	1.e-2	1000	1.0	161.89	16250	1763114	19	2283
50	1.e-2	1000	0.5	139.68	16227	1763162	19	1602
75	1.e-2	1000	0.5	157.37	16227	1763162	19	2147
100	1.e-2	1000	0.5	163.20	16227	1763162	19	2289
50	1.e-2	1000	0.25	142.30	10652	1833172	19	1475
100	1.e-2	1000	0.25	153.51	10652	1833172	19	1782
50	1.e-2	1000	0.1	135.54	4167	1881569	19	982
75	1.e-2	1000	0.1	133.64	4167	1881569	19	982
100	1.e-2	1000	0.1	133.13	4167	1881569	19	982

preconditioners.

REFERENCES

[1] E. D. ANDERSEN, J. GONDZIO, C. MÉSZÁROS, AND X. XU, *Implementation of interior point methods for large scale linear programming*, in Interior Point Methods in Mathematical Programming, T. Terlaky, ed., Kluwer Academic Publishers, 1996, pp. 189–252.

[2] M. BENZI, G. GOLUB, AND J. LIESEN, *Numerical solution of saddle point problems*, Acta Numerica, 14 (2005), pp. 1–137.

[3] M. BENZI AND V. SIMONCINI, *On the eigenvalues of a class of saddle point matrices*, Numerische Mathematik, 103 (2006), pp. 173–196.

[4] 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.

[5] T. COLEMAN AND A. VERMA, *A preconditioned conjugate gradient approach to linear equality constrained minimization*, Computational Optimization and Applications, 20 (2001), pp. 61–72.

[6] H. DOLLAR AND A. WATHEN, *Approximate factorization constraint preconditioners for saddle-point matrices*, SIAM Journal on Scientific Computing, 27 (2005), pp. 1555–1572.

[7] R. W. FREUND AND N. M. NACHTIGAL, *Software for simplified Lanczos and QMR algorithms*, Applied Numerical Math-

ematics, 19 (1995), pp. 319–341.
[8] 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.
[9] 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.
[10] 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.
[11] 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.
[12] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1997.