Some Preconditioned Conjugate Gradient Algorithms for the Solution of Equality Constrained Quadratic Programming Problems

Silvia Bonettini

Dipartimento di Matematica, Università di Ferrara
via Saragat 1, Blocco B, 44100 Ferrara, Italy
E-mail: bntslv@unife.it
www.unife.it

In this paper we consider the application of the preconditioned conjugate gradient (PCG) method to the solution of equality constrained quadratic programming problems. In particular, we consider three different PCG algorithms and two indefinite preconditioners. A special attention is given to the choice of the factorization method for the preconditioner. The numerical experiments show a comparison of the effectiveness of the proposed variants. Furthermore, we show the behaviour of the PCG method for the solution of the inner subproblem arising at each step of an interior point algorithm for the solution of non linear programming problems.

Keywords: Preconditioned Conjugate Gradient Method, Indefinite Preconditioners, Large Scale Optimization, Nonlinear Programming Problems.

1. Introduction

This work is concerned with the solution of the quadratic programming problem

$$\begin{array}{l} \min \quad \frac{1}{2}x^T H x - c^T x \\ Ax = b \end{array} \tag{1}$$

where H is an $n \times n$ symmetric matrix, while the matrix A is $m \times n$.
The Karush–Kuhn–Tucker (KKT) conditions of the problem (1) are represented by the system

$$\begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c \\ b \end{pmatrix}, \tag{2}$$

whose coefficient matrix is the following indefinite $n + m$ matrix

$$M = \begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix}. \tag{3}$$

It is well known that a sufficient condition for the nonsingularity of (3) is that

(A1) A is full row rank and H is positive definite on the null space of A, which means $p^T Hp > 0$ for any $p \in \mathbb{R}^n$ $(p \neq 0)$ such that $Ap = 0$.

We are motivated to study the numerical solution of the problem (1) (or, equivalently, (2)) since in the framework of the interior point approach, a variety of algorithms for linearly and nonlinearly constrained optimization (see, for example, Refs. 1–5) requires, at each step, the solution of a subproblem of such form.

Recently, many authors propose as efficient iterative linear solver for the system (2), the preconditioned conjugate gradient (PCG) method, with an indefinite preconditioner having the same block structure of the matrix (3)

$$P = \begin{pmatrix} G & A^T \\ A & 0 \end{pmatrix}, \tag{4}$$

where G is a positive definite approximation of H.
In Ref. 6, under suitable hypotheses, the authors prove that in exact arithmetic the PCG method with such preconditioner applied to the system (2) terminates in a finite number of steps, and they provide also a spectral analysis of the matrix MP^{-1}. The same preconditioner and its variants have been further investigated (see for example Refs. 4,7–9 and references therein).
The matrix P has a very special structure, which yields important properties. Let us consider the system

$$\begin{pmatrix} G & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \bar{g} \\ \underline{g} \end{pmatrix} = \begin{pmatrix} \bar{r} \\ 0 \end{pmatrix}. \tag{5}$$

where, here and in the following, we indicate with $\bar{g}$ the n the first n components and with $\underline{g}$ the last m components of the $n + m$ vector g. If the vector g solves the previous system, then the component $\bar{g}$ is the projection of r in the null space of the matrix A.
More precisely, we have

$$\bar{g} = G^{-1}(r - A^T \underline{g}) = (G^{-1} - G^{-1}A^T(AG^{-1}A^T)^{-1}AG^{-1})r = P_A r$$

where we denote by P_A the projection operator on the null space of A

$$P_A = G^{-1} - G^{-1}A^T(AG^{-1}A^T)^{-1}AG^{-1}. \tag{6}$$

This property plays a crucial role in the analysis of the PCG method made in Refs. 6,10,11.

In the next section we recall the main theoretical results about the PCG method applied to the problem (2) with the preconditioner P, and we describe three different variants of the method, focusing on the instability that the finite precision could produce.

In section 3, we consider the following variant of the preconditioner P,

$$\tilde{P} = \begin{pmatrix} G & A^T \\ A & -E \end{pmatrix}, \tag{7}$$

where E is a positive definite diagonal $m \times m$ matrix.

In section 4, we compare the effectiveness of the PCG algorithms described in the sections 3 and 4 on a set of equality constrained quadratic programs. Furthermore, we employed the PCG method as inner solver for the system arising at each step of the interior point algorithm described in Ref. 12, comparing the performance of the whole interior point algorithm with respect to the different algorithms.

2. The preconditioned conjugate gradient method

By putting

$$v = \begin{pmatrix} x \\ y \end{pmatrix}, \; z = \begin{pmatrix} c \\ b \end{pmatrix},$$

the system (2) can be written as $Mv = z$. The PCG method applied to (2) can be written as follows:

Algorithm 2.1. *Choose an initial point v_0, compute $r_0 = z - Mv_0$, $g_0 = P^{-1}r_0$, $\nu = {r_0}^T g_0$ and put $p_0 = g_0$;*
for $i = 0, 1, \ldots$ until a stopping criterion is satisfied

$$\delta \leftarrow {p_i}^T M p_i \tag{8}$$
$$\alpha \leftarrow \nu/\delta \tag{9}$$
$$r_{i+1} = r_i - \alpha M p_i \tag{10}$$
$$v_{i+1} = v_i + \alpha p_i \tag{11}$$
$$g_{i+1} = P^{-1} r_{i+1} \tag{12}$$
$$\delta \leftarrow \nu \tag{13}$$
$$\nu \leftarrow {r_{i+1}}^T g_{i+1} \tag{14}$$
$$\beta \leftarrow \nu/\delta \tag{15}$$
$$p_{i+1} = g_{i+1} + \beta p_i \tag{16}$$

Since M is an indefinite matrix, it can happen that, for some index i, the quantity δ computed at the step (8) is zero: in this case, we say that a *breakdown* occurs for the algorithm.
It is possible to prove that if the starting point is chosen such that $Ax_0 = b$, then the points x_i generated by the PCG procedure are such that $Ax_i = b$, then the linear system which has to be solved at the step (12) has the form (5) at each iterate i (see Theorem 2.1 in Ref. 10).
Thus, the step (12) is equivalent to a projection of the vector $\bar{r}_i$ on the null space of the matrix A. This yields very strong properties to the Algorithm 2.1, which can be stated as follows (for the proof, see Theorem 3.5 in Ref. 6 and Theorem 2.2 in Ref. 10):

Theorem 2.1. *Let us assume that the hypothesis A1 holds and let Z be a matrix whose columns form a basis for the null space of A, so that every vector $u \in \mathbb{R}^n$ can be written as $u = Zu^\tau + A^T u^\nu$.*
Let x_* the first n components of the solution of the system (2). If we choose a starting point v_0 such that $Ax_0 = b$, then the components $\{x_k^\tau\}$ of the elements of the sequence $\{x_k\}$ generated by the Algorithm 2.1 applied to the system (2) are the elements of the sequence generated by the conjugate gradient method with the preconditioner $Z^T GZ$ applied to the system

$$Z^T HZx^\tau = Z^T c_z \tag{17}$$

where $c_z = c - HA^T x_*^\nu$.
Furthermore, the Algorithm 2.1 does not break down and it finds the solution x_* after at most $n - m$ iterations, and for $1 \leq i \leq n - m$ the following estimation holds

$$\|x_i - x_*\| \leq 2\sqrt{k} \left(\frac{1 - \sqrt{k}}{1 + \sqrt{k}} \right)^i \|x_0 - x_*\|, \tag{18}$$

where $k = k(Z^T HZ(Z^T GZ)^{-1})$ and $\|\cdot\|$ is the euclidean norm in $\mathbb{R}^n$.

2.1. *Two projection algorithms*

The Algorithm 2.1 actually solves the problem

$$\min \frac{1}{2} x^{\tau T} Z^T HZx^\tau - x^{\tau T} c_Z$$

that we obtain by substituting the expression $x = Zx^\tau + A^T x^\nu$ in (1). A similar approach is used in the Algorithm 2 in Ref. 11, where the PCG iteration is applied only to the primal variable x.

In this case, the residual vector is defined as $\tilde{r}^+ = Hx^i - c$. The residual $\tilde{r}_i^+$ in the algorithm, will be bounded away from zero, but, as the iterates approach to the solution, it will become increasingly closer to the range of A^T. Indeed, if $(x_*^T, y_*^T)^T$ is the solution of the system (2), we have $c - Hx_* = A^T y_*$. Thus, the projection of the residual on the null space of A, the vector $\tilde{g}_i$, will become increasingly closer to zero.
This difference in the magnitudes of $\tilde{r}_i^+$ and $\tilde{g}_i$ might cause numerical difficulties, as shown in Example 1 in Ref. 11, and it leads the projected residual to do not belong exactly to the null space of A.
In order to avoid this drawback, in Ref. 11 the authors propose a variant of the PCG algorithm, which, at each step, provides a least squares estimate of the component of the residual in the range space of A^T, which is orthogonal to the null space of A; then, the residual is updated by subtracting this component. This update leads the revised residual $\tilde{r}_i$ to become increasingly closer to zero as the iterates approach to the solution.

Algorithm 2.2 ([11],Algorithm III). *Choose an initial point x_0 such that $Ax_0 = b$ and compute $\tilde{r}_0^+ = Hx_0 - c$, $v_0 = argmin\|\tilde{r}_0^+ - A^T v\|_{G^{-1}}$, $\tilde{r}_0 = \tilde{r}_0^+ - A^T v_0$, $\tilde{g}_0 = P_A \tilde{r}_0$, and put $\tilde{p}_0 = \tilde{g}_0$;*
for $i = 0, 1, ...$ until a stopping criterion is satisfied

$$
\begin{aligned}
\alpha &\leftarrow \frac{\tilde{r}_i^T \tilde{g}_i}{\tilde{p}_i^T H \tilde{p}_i} \\
x_{i+1} &= x_i + \alpha \tilde{p}_i \\
\tilde{r}_{i+1}^+ &= \tilde{r}_i^+ + \alpha H \tilde{p}_i \\
v_{i+1} &= argmin\|\tilde{r}_{i+1}^+ - A^T v\|_{G^{-1}} \qquad (19)\\
\tilde{r}_{i+1} &= \tilde{r}_{i+1}^+ - A^T v_{i+1} \qquad (20)\\
\tilde{g}_{i+1} &= P_A \tilde{r}_{i+1} \qquad (21)\\
\beta &\leftarrow \frac{\tilde{r}_{i+1}^T p_{i+1}}{\tilde{r}_i^T \tilde{g}_i} \\
\tilde{p}_{i+1} &= \tilde{g}_{i+1} + \beta \tilde{p}_i
\end{aligned}
$$

Here and in the following we denote by $\|u\|_{G^{-1}}$ the norm $\sqrt{u^T G^{-1} u}$, with G^{-1} positive definite.
In order to compute the projection (21), we have to solve two systems: first,

we obtain v_{i+1} in (19) by solving the system

$$\begin{pmatrix} G & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} g \\ v_{i+1} \end{pmatrix} = \begin{pmatrix} \tilde{r}_{i+1}^+ \\ 0 \end{pmatrix}$$

and then, to obtain the vector $\tilde{g}_{i+1}$ in (21), we solve

$$\begin{pmatrix} G & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \tilde{g}_{i+1} \\ u \end{pmatrix} = \begin{pmatrix} \tilde{r}_{i+1}^+ - A^T v_{i+1} \\ 0 \end{pmatrix}.$$

In exact arithmetic the vector $\tilde{r} = \tilde{r}^+ - A^T v$ computed at the step (20) belongs to the null space of A, thus the step (21) can be considered as an iterative refinement step.
Furthermore, we can obtain an estimate of the Lagrange multiplier as $y_i = -\sum_{k=0}^{i} v_i$ and, then the vector $\tilde{r}_i$ represents the residual of the first equation of the system (2): indeed we have $\tilde{r}_i = Hx_i + A^T y_i - c$.
In the Algorithm 2.1 the vector g^{i+1} is obtained by solving one system only

$$\begin{pmatrix} G & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \bar{g}_{i+1} \\ \underline{g}_{i+1} \end{pmatrix} = \begin{pmatrix} \bar{r}_{i+1} \\ \underline{r}_{i+1} \end{pmatrix},$$

where $\bar{r}_{i+1} = c - Hx_{i+1} - A^T y_{i+1}$ and $\underline{r}_{i+1} = b - Ax_{i+1}$.
In exact arithmetic, the component $\underline{r}_{i+1}$ of the residual should be the null vector, but operating in finite arithmetic this is not guaranteed.

We propose a further variant of the PCG method, which can be obtained by projecting the residual $\hat{r}_i^+ = c - Hx_i$ on the null space of A and providing a least squares estimate of the multiplier y_i at each step.
The resulting algorithm can be written as follows:

Algorithm 2.3. *Choose an initial point x_0 such that $Ax_0 = b$ and compute $\hat{r}_0^+ = c - Hx_0$, $y_0 = argmin\|\hat{r}_0^+ - A^T y\|_{G^{-1}}$, $\hat{r}_0 = \hat{r}_0^+ - A^T y_0$, $\hat{g}_0 = P_A \hat{r}_0$, and put $\hat{p}_0 = \hat{g}_0$;*

for $i = 0, 1, ...$ *until a stopping criterion is satisfied*

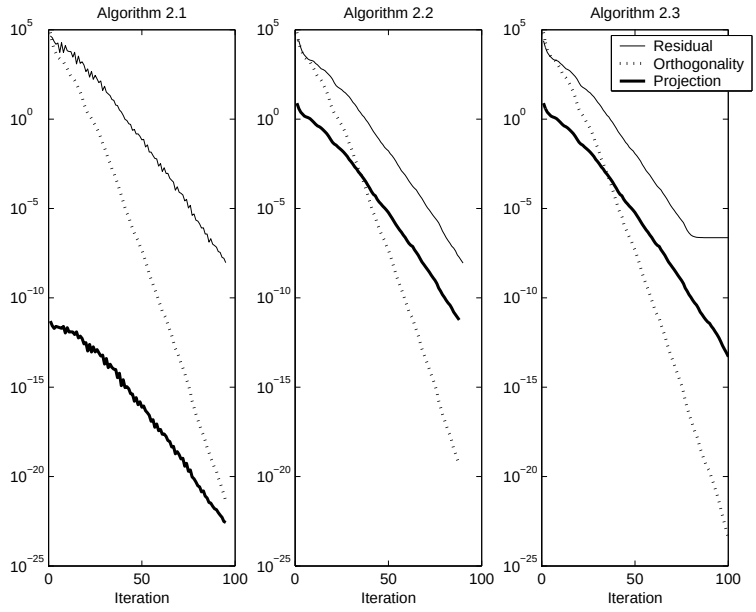
$$\begin{aligned}
\alpha &\leftarrow \frac{\hat{r}_i^T \hat{g}_i}{\hat{p}_i^T H \hat{p}_i} \\
x_{i+1} &= x_i + \alpha \hat{p}_i \\
\hat{r}_{i+1}^+ &= \hat{r}_i^+ - \alpha H \hat{p}_i \\
y_{i+1} &= argmin \|\hat{r}_{i+1}^+ - A^T y\|_{G^{-1}} \\
\hat{r}_{i+1} &= \hat{r}_{i+1}^+ - A^T y_{i+1} \\
\hat{g}_{i+1} &= P_A \hat{r}_{i+1} \\
\beta &\leftarrow \frac{\hat{r}_{i+1}^T p_{i+1}}{\hat{r}_i^T \hat{g}_i} \\
\hat{p}_{i+1} &= \hat{g}_{i+1} + \beta \hat{p}_i
\end{aligned}$$

It is interesting to observe the effects of the finite precision on the Algorithms 2.1, 2.2 and 2.3.
The figure 2.1 shows a comparison between the Algorithms 2.1, 2.2 and 2.3 on the test problem CVXEQP3 of the CUTE collection [13] with $n = 1000$ and $m = 750$; the matrix P has a condition number of order 10^{12}.
The solution of the systems involved in the three algorithms is computed by means of the direct factorization of the preconditioner P obtained with the MA27 routine of the Harwell Subroutine Library [14].
For each iteration i, we have considered the following quantities: the norm of the residuals r_i, $\tilde{r}_i$ and $\hat{r}_i$ which indicate the progress towards the solution of the system (Residuals); the scalar products ${r_i}^T g_i$, $\tilde{r}_i^T \tilde{g}_i$ and $\hat{r}_i^T \hat{g}_i$, which are measurements of the angle between the residuals and g_i, $\tilde{g}_i$ and $\hat{g}_i$ respectively (Orthogonality); the quantities $\|A\bar{g}_i\|$, $\|A\tilde{g}_i\|$ and $\|A\hat{g}_i\|$, which tells us how precisely the projection is computed (Projection).
After 100 iterations, we can observe that for the Algorithm 2.1, the projection is computed with a residual whose norm is less than 10^{-20}, while for the algorithms 2.2 and 2.3, the projection is computed with a residual greater than 10^{-15}.
Furthermore, we observe that norm of the residual $\|\hat{r}_i\|$ in the Algorithm 2.3 is greater than 10^{-7}
Thus, the finite precision does not significantly influence the Algorithm 2.1, while it leads the algorithms 2.2 and 2.3 to have less accuracy in the results.

3. A different preconditioner

In this section we consider the Algorithm 2.1 with the preconditioner $\tilde{P}$ defined as in (7).

Fig. 1. Test problem CVXEQP3



In this case, the Theorem 2.1 can not apply, but the finite termination property is ensured by the Theorem 3.4 in Ref. 6, which can be stated as follows.

Theorem 3.1. *Consider the PCG method applied to the system $Mv = k$ with preconditioner P, where M and P are two $n \times n$ symmetric nonsingular matrices. If a breakdown does not occur, then the Algorithm 2.1 finds the solution of the system in at most l iterates, where l is the dimension of the Krylov space $\mathcal{K} = Span\{r_0, MP^{-1}r_0, (MP^{-1})^2 r_0, ...\}$.*

We can prove the previous theorem using the same arguments employed in the proof of the Theorem 3.4 in Ref. 6, part (a).
If M is defined as in (3) and employing the preconditioner $\tilde{P}$ defined (7), under the hypothesis A1 the previous theorem holds.
Even if the preconditioner $\tilde{P}$ has weaker theoretical properties than P, it is very interesting from the numerical point of view.
Indeed, the matrix P admits a Cholesky–like factorization of the form LDL^T where L is a lower triangular matrix with diagonal entries equal to one and D is a nonsingular diagonal matrix with n positive and m negative diagonal entries. In order to reduce the fill–ins in the lower triangular

factor, we can perform a minimum degree reordering of the matrix P, but it is not assured that the permuted matrix can be factorized in the Cholesky–like form.
Nevertheless, we can obtain a factorization in the form LDL^T if we use for the matrix P the regularization technique described in Ref. 15: if a pivot d_i is too small ($|d_i| < 10^{-15} \max_{j<i} |d_j|$), we put $d_i = \sqrt{\epsilon}$ if $1 \leq i \leq n$, or $d_i = -\sqrt{\epsilon}$ if $n+1 \leq i \leq n+m$, where ϵ is the machine precision.
By applying the regularization technique during the factorization of P, we actually obtain the factorization of $\tilde{P}$.
The Cholesky–like factorization can be implemented by modifying the Ng and Peyton package as described in Ref. 16. This new package, called BLKFCLT and downloadable from the web page *http://dm.unife.it/blkfclt/*, is structured in two phases: the first phase provides an *a priori* reordering routine for the sparsity preserving and the computation of a symbolic factorization, while, in the second phase, the Cholesky–like factorization is computed, employing the dynamic regularization strategy.

4. Numerical results

The aim of our numerical experience is to compare the effectiveness of the different PCG algorithms as solvers for equality constrained quadratic programs. Furthermore, we considered the PCG method as inner solver in the inexact Newton interior point algorithm described in Ref. 12, which at each step has to solve a system of the form (2).
The numerical results presented in the following have been carried out by coding the algorithms in C++, on a HP zx6000 workstation with Itanium2 processor 1.3 GHz and 2 Gb of RAM; the code is provided of an AMPL interface.
The first comparison in table 1 shows the performances of the PCG algorithms on the solution of the system arising at the last iterate of the interior point method applied to the nonlinear programming problems listed in the first column of the table.
The starting point is the same in all cases. The stopping criterion for the PCG procedure is $\|r_i\| \leq 10^{-8}$ for the Algorithm 2.1, $\|\tilde{r}_i\| \leq 10^{-8}$ for the Algorithm 2.2 and $\|\hat{r}_i\| \leq 10^{-8}$ for the Algorithm 2.3.
The table reports the dimension of the system (n and m), the iterations number and, in brackets, the total execution time in seconds, that is the factorization plus the solution time. The symbol ' * ' indicates that the tolerance of 10^{-8} was not satisfied after $n+2$ iterations.
We can observe that the Algorithm 2.1 with the preconditioner (7) gives

the best results in terms of time: this is due also to the effectiveness of the Cholesky–like factorization subroutine BLKFCLT described in the previous section. Furthermore, the Algorithm 2.1 solves one system only at each step, while the other algorithms has to solve two systems.
The table 2 contains the comparison of four different versions of the interior point method on a set of nonlinear programming problems. Each version is obtained by applying a different PCG algorithm to the solution of the inner linear system.
In this case, the termination criterion for the PCG procedure exploits an adaptive stopping rule which depends on the violation of the KKT optimality conditions at the current iterate of the interior point algorithm.
In the table, we report the execution time (in seconds) of the interior point algorithm and the number of iterations: the first number in the columns 'iter' is referred to the outer iterations, while in brackets we indicate the total number of PCG iterations.
We declare a failure of the interior point algorithm if the line–search strategy requires more than 10 backtracking reductions (see Ref. 12) or when the factorization routine requires an excessive memory storage.
We can see that the versions of the interior point method with the algorithms 2.2 and 2.3 as inner solvers fail on 2 and 3 problems respectively, while, employing the Algorithm 2.1 we observe one only failure: this failure occurs when the preconditioner (4) is factorized by means of the MA27 routine and it is due to a lack of memory.
Furthermore, in the most part of the cases, the best results in terms of time are obtained by the version which employs the Algorithm 2.1 with the preconditioner (7), factorized with the BLKFCLT routine. In some cases, for example for the test problems 'marine', 'pinene', 'optcdeg2', the reduction of the execution time is remarkable.

Acknowledgements

This research was supported by the Italian Ministry for Education, University and Research (MIUR), FIRB Project RBAU01JYPN.

References

1. A. Wächter and L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Mathematical Programming*, **106**, 25 – 57 (2006).

Table 1. Comparison of the PCG algorithms with the preconditioners (4) and (7)

Problem	Ref.	n	m	Algorithm 2.1[1]	Algorithm 2.2	Algorithm 2.3	Algorithm 2.1[2]
catmix	[17]	2198	1598	200 (1.95)	186 (2.58)	146 (2.2)	200 (0.35)
channel	[17]	6398	6398	0 (0.07)	0 (0.07)	1 (0.07)	0 (0.04)
dtoc6	[13]	10000	5000	1 (0.06)	1 (0.06)	1 (0.06)	1 (0.02)
gouldqp2	[13]	699	349	138 (0.04)	135 (0.05)	135 (0.04)	138 (0.04)
marine	[17]	6415	6352	37 (11.8)	62 (12.2)	*	4 (0.08)
optcdeg2	[17]	11998	7999	94 (10.7)	91 (12.7)	96 (12.9)	87 (0.55)
optcdeg3	[13]	11998	7999	134 (22.6)	127 (26.8)	127 (26.7)	120 (0.75)
pinene	[17]	2000	1995	3 (12.1)	0 (12)	1 (12.1)	3 (0.02)
robot	[17]	14398	9600	3 (0.33)	2 (0.33)	2 (0.32)	3 (0.24)
steering	[17]	3999	3200	1 (11.8)	0 (11.8)	1 (11.8)	1 (0.03)

Note: [1] with preconditioner (4); [2] with preconditioner (7); $*$ can not reach the required tolerance

Table 2. Comparison of the PCG algorithms as inner solver in an interior point method

Problem	Algorithm 2.1[1]		Algorithm 2.2		Algorithm 2.3		Algorithm 2.1[2]	
	time	iter	time	iter	time	iter	time	iter
catmix	111.4	253(15653)	16.7	120(965)	121.7	308(2655)	20.17	218(9955)
channel	1.43	4(137)	0.58	4(0)	0.6	4(4)	0.63	4(19)
dtoc6	11.1	71(384)	11.4	71(320)	11.3	71(320)	11.1	71(385)
gouldqp2	0.25	21(1023)	0.32	21(1017)	0.32	21(1016)	0.26	21(1023)
marine	m	m	m	m	m	m	2.32	29(67)
optcdeg2	190.1	53(576)	205.2	53(564)	206.1	53(580)	5.84	53(590)
optcdeg3	544.2	48(581)	562.9	48(559)	563.1	48(561)	5.5	48(541)
pinene	450.32	38(78)	510.7	43(11)	451.2	38(39)	1.42	39(89)
robot	11.2	39(127)	*	*	*	*	3.87	35(95)
steering	177.7	34(215)	154.1	25(37)	*	*	13.9	33(203)

Note: [1] with preconditioner (4); [2] with preconditioner (7); $*$ failure of the interior–point algorithm; m not enough memory for requirements of the factorization routine.

2. R. J. Vanderbei and D.F. Shanno, An interior-point algorithm for nonconvex nonlinear programming, *Computational Optimization and Applications*, **13**, 231–252 (1999).
3. R.A.Waltz, J.L. Morales, J. Nocedal and D. Orban, An interior point algorithm for nonlinear optimization that combines line search and trust region steps, *Mathematical Programming*, **107**, 391 – 408 (2006).
4. L. Bergamaschi, J. Gondzio and G. Zilli, Preconditioning indefinite systems in interior point methods for optimization, *Computational Optimization and Applications*, **28**, 149–171 (2004).
5. R. H. Byrd, J. C. Gilbert and J. Nocedal, A trust region method based on interior point techniques for nonlinear programming, *Mathematical Programming*, **89**, 149–185 (2000).
6. L. Lukšan and J. Vlček, Indefinitely preconditioned truncated Newton method for large sparse equality constrained nonlinear programming problems, *Numer-*

ical Linear Algebra with Applications, **5**, 219–247 (1998).
7. L. Lukšan, C. Matonoha and J. Vlček, Interior–point method for nonlinear nonconvex optimization, *Numerical Linear Algebra with Applications*, **11**, 431–453 (2004).
8. C. Durazzi and V. Ruggiero, *Indefinitely preconditioned conjugate gradient method for large sparse equality and inequality constrained quadratic problems*, *Numer. Linear Algebra Appl.*, **10**, 673-688 (2003).
9. S. Cafieri, M. D'Apuzzo,V. De Simone and D. di Serafino, On the iterative solution of KKT systems in potential reduction software for large scale quadratic problems, *Computational Optimization and Applications*, to appear (2004).
10. S. Bonettini, V. Ruggiero and F. Tinti, On the Solution of Indefinite Systems Arising in Nonlinear Optimization, Technical report n. 359, Dipartimento di Matematica, Università di Ferrara (2006).
11. N.I.M. Gould, M. E. Hribar, and J. Nocedal, On the solution of equality constrained quadratic programming problems arising in optimization, *SIAM J. Sci. Comput.*, **23**, 1376–1395 (2001).
12. S. Bonettini, E. Galligani and V. Ruggiero, Inner solvers for interior point methods for large scale nonlinear programming, *Computational Optimization and Applizations*, to appear (2005).
13. I. Bongartz, A.R. Conn, N. Gould and Ph. L. Toint, *CUTE: Constrained and Unconstrained Testing Environnment*, ACM Transactions on Mathematical Software, **21**, 123–160 (1995).
14. I.S. Duff and J.K. Reid, A set of Fortran subroutines for solving sparse symmetric sets of linear equations. Tech. Report AERE R10533, HMSO, London, (1982).
15. A. Altman and J. Gondzio, Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization, *Optim. Methods Softw.*, **11–12**, 275–302 (1999).
16. S. Bonettini and V. Ruggiero, Some iterative methods for the solution of a reduced symmetric indefinite KKT system, *Computational Optimization and Applizations*, to appear (2005).
17. E.D. Dolan, J. J. Moré and T. S. Munson, *Benchmarking optimization software with COPS 3.0*, Technical Report ANL/MCS-TM-273, Argonne National Laboratory, Illinois, USA (2004).