On the Solution of Indefinite Systems Arising in Nonlinear Optimization [*]

Silvia Bonettini, Valeria Ruggiero and Federica Tinti
Dipartimento di Matematica, Università di Ferrara

Abstract

We consider the application of the preconditioned conjugate gradient (PCG) method to the solution of indefinite linear systems arising in nonlinear optimization. Our approach is based on the choice of quasidefinite preconditioners and of a suitable factorization routine. Some theoretical and numerical results about these preconditioners are obtained. Furthermore, we show the behaviour of the PCG method for different representations of the indefinite systems and we compare the effectiveness of the proposed variants.

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

1 Introduction

This work is concerned with the solution of an indefinite linear system whose coefficient matrix has the following form:

$$M = \begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix} \tag{1}$$

where H is an $\bar{n} \times \bar{n}$ symmetric matrix, while the matrix A is $\bar{m} \times \bar{n}$. This system is related to the first order Karush–Kuhn–Tucker (KKT) conditions of the following quadratic programming problem

$$\begin{array}{cl} \min & \frac{1}{2}x^T Hx - c^T x \\ Ax = b & \end{array} \tag{2}$$

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

which can be expressed as follows:

$$\begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c \\ b \end{pmatrix}. \tag{3}$$

It is well known that a sufficient condition for the nonsingularity of (1) 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 $p \in \mathbb{R}^{\bar{n}}$ $(p \neq 0)$ such that $Ap = 0$.

The numerical solution of systems as (3) is required also in nonlinear programming problems. Indeed, in the framework of the interior–point methods, a variety of algorithms for linearly and nonlinearly constrained optimization (see, for example, [20], [21], [19], [4], [6]) requires, at each step, the solution of the Newton system (or of a perturbation to it) applied to the KKT optimality conditions of the problem:

$$\begin{array}{ll} \min & f(x) \\ h(x) = 0 & \\ g(x) \geq 0 & \end{array} \tag{4}$$

where $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, $h(x) : \mathbb{R}^n \rightarrow \mathbb{R}^{neq}$, $g(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ are twice continuously differentiable functions. The inequality constraints are often reformulated by introducing the vector of the slack variables s as

$$\begin{array}{l} g(x) - s = 0 \\ s \geq 0. \end{array} \tag{5}$$

In this case, the coefficient matrix of the Newton's system is

$$\begin{pmatrix} Q & B^T & C^T & 0 \\ B & 0 & 0 & 0 \\ C & 0 & 0 & I \\ 0 & 0 & S & W \end{pmatrix}, \tag{6}$$

with $Q = \nabla^2 f(x) - \sum_1^{neq} \lambda_i \nabla^2 h_i(x) - \sum_1^m w_i \nabla^2 g(x)$, $B = -\nabla h_i(x)^T$ and $C = -\nabla g_i(x)^T$. Here Q is the Hessian matrix of the Lagrangian function of the problem (4), $\nabla^2 f(x)$, $\nabla^2 h_i(x)$, $\nabla^2 g_i(x)$ are the Hessian matrices of the function $f(x)$ and of the i–th component of the constraints $h(x)$ and $g(x)$ respectively. Furthermore, $\lambda \in \mathbb{R}^{neq}$ and $w \in \mathbb{R}^m$ are the Lagrange multipliers of the equality and inequality constraints respectively.

The matrices S and W are diagonal matrices whose entries are the components of the vectors s and w respectively. Since in the interior–point approach the slack variables and the multipliers related to the inequalities are forced to be positive, we will assume that the diagonal entries of the matrices S and W are positive.
The matrix (6) is not symmetric; thus it is usual to obtain a symmetric representation of the Newton system by a suitable scaling of the equations and by applying elimination techniques.
From the literature, we devised four different representations of the Newton system, which will be described in the section 3. The coefficient matrices of these reformulations of the Newton system have the same block structure (1).
Recently, many authors propose as efficient iterative linear solver for the system (3), the preconditioned conjugate gradient (PCG) method, with an indefinite preconditioner with the same block structure of the matrix (1)

$$P = \left(\begin{array}{cc} G & A^T \\ A & 0 \end{array} \right), \tag{7}$$

where G is a positive definite approximation of H.
In [14], under suitable hypotheses, the authors prove that the PCG method with such preconditioner applied to the system (3) terminates in a finite number of steps in exact arithmetic, providing also a spectral analysis of the matrix MP^{-1}. The same preconditioner and its variants have been further investigated (see for example [4], [15], [11], [7] and references therein).
The matrix P has a very special structure, which yields important properties. Indeed, if we consider the augmented system related to the first order condition of the least squares problem $\min_{\underline{g}} \|r - A^T\underline{g}\|^2_{G^{-1}}$, where $\|u\|^2_{G^{-1}} = u^T G^{-1} u$ (G^{-1} positive definite), we obtain

$$\left(\begin{array}{cc} G & A^T \\ A & 0 \end{array} \right) \left(\begin{array}{c} \bar{g} \\ \underline{g} \end{array} \right) = \left(\begin{array}{c} r \\ 0 \end{array} \right). \tag{8}$$

Here and in the following, if g is a $\bar{n} + \bar{m}$ vector, we indicate with $\bar{g}$ the $\bar{n}$ vector whose entries are the first $\bar{n}$ components of g and with $\underline{g}$ the $\bar{m}$ vector whose entries are the last $\bar{m}$ components of g, so that

$$g = \left(\begin{array}{c} \bar{g} \\ \underline{g} \end{array} \right).$$

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

$$P_A = G^{-1} - G^{-1}A^T(AG^{-1}A^T)^{-1}AG^{-1}. \tag{9}$$

This implies $A\bar{g} = 0$.
This property plays a crucial role in the analysis of the PCG method that enables us to derive in a more general way the theoretical features of the PCG method, reported in the next section. We focus on the connection between the preconditioning technique with the matrix P and the projection operation on the null space of the matrix A.
Moreover, we show that the approach followed in [14] and the one suggested in [12], which provides to solve the quadratic problem (2) by projecting at each step the current residual on the null space of A, are very similar. Nevertheless, the first approach could prevent the numerical instability problems which may arise following the second approach.
In section 4, a set of numerical experiments enables us to compare the effectiveness of the PCG method for the solution of the Newton system in four different formulations. Furthermore, we employed the PCG method as inner solver for the Newton system arising at each step of the interior point algorithm described in [2], comparing the performance of the whole interior point algorithm with respect to the different representations of the inner linear system.

2 The preconditioned conjugate gradient method

By putting

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

the system (3) can be written as $Mv = k$. The PCG method applied to (3) can be written as follows:

Algorithm 2.1 *Choose an initial point v_0, compute $r_0 = k - Mv_0$, $g_0 = P^{-1}r_0$, $\nu = {r_0}^T g_0$ and put $p_0 = g_0$;*

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

$$\begin{aligned}
\delta &\leftarrow {p_i}^T M p_i && (10)\\
\alpha &\leftarrow \nu/\delta && (11)\\
r_{i+1} &= r_i - \alpha M p_i && (12)\\
v_{i+1} &= v_i + \alpha p_i && (13)\\
g_{i+1} &= P^{-1} r_{i+1} && (14)\\
\delta &\leftarrow \nu && (15)\\
\nu &\leftarrow {r_{i+1}}^T g_{i+1} && (16)\\
\beta &\leftarrow \nu/\delta && (17)\\
p_{i+1} &= g_{i+1} + \beta p_i && (18)
\end{aligned}$$

Since M is an indefinite matrix, it can happen that, for some index i, the quantity δ computed at the step (10) is zero: in this case, we say that a *breakdown* occurs for the algorithm. It can be proved (Theorem 3.4 in [14]) that, if P is defined as in (7), A is full row rank, M is a symmetric nonsingular matrix and a breakdown does not occur, then, starting from any $v_0 \in \mathbb{R}^{\bar{n}+\bar{m}}$, the Algorithm 2.1 finds a solution of the system $Mv = k$ in at most $\bar{n} - \bar{m} + 2$ iterations.

The approach described above can be considered a "primal–dual" approach to the solution of the programming problem (2), since the Algorithm 2.1 is equivalent to the PCG method applied to the determination of a stationary point of the Lagrangian function of the problem (2)

$$\mathcal{L}(x, y) = \frac{1}{2} x^T H x - c^T x + y^T (Ax - b).$$

Indeed, at each step (12), we have

$$r_i = -\begin{pmatrix} \nabla_x \mathcal{L}(x_i, y_i) \\ \nabla_y \mathcal{L}(x_i, y_i) \end{pmatrix}.$$

2.1 Preconditioning techniques and projection procedures

The next theorem shows that, if the starting point of the PCG procedure is chosen so that $Ax_0 = b$, then the linear system which has to be solved at the step (14) has the form (8) at each iterate i.

Theorem 2.1 *Assume that P defined as in (7), A is full row rank and M is a symmetric nonsingular matrix.*

If the starting point

$$v_0 = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \tag{19}$$

is chosen such that x_0 solves the system $Ax = b$ and a breakdown does not occur, then the Algorithm 2.1 generates a sequence of points x_i such that $Ax_i = b$ so that

$$r_i = \begin{pmatrix} \bar{r}_i \\ 0 \end{pmatrix}.$$

Moreover, the direction p is given by

$$p_i = \begin{pmatrix} \bar{p}_i \\ \underline{p}_i \end{pmatrix},$$

where $\bar{p}_i$ belongs to the null space of A.

Proof. The inverse of the preconditioner P is given by

$$P^{-1} = \begin{pmatrix} G^{-1} - G^{-1}A^T(AG^{-1}A^T)^{-1}AG^{-1} & G^{-1}A^T(AG^{-1}A^T)^{-1} \\ (AG^{-1}A^T)^{-1}AG^{-1} & -(AG^{-1}A^T) \end{pmatrix}. \tag{20}$$

The left up block of the matrix P^{-1} is the projection operator on the null space of A defined in (9).
If the starting point (19) is such that x_0 is a solution of the system $Ax = b$, then we have

$$r_0 = \begin{pmatrix} c - Hx_0 - A^T y_0 \\ 0 \end{pmatrix} = \begin{pmatrix} \bar{r}_0 \\ 0 \end{pmatrix}.$$

Then, when we compute $g_0 = P^{-1}r_0$, we have

$$g_0 = \begin{pmatrix} P_A \bar{r}_0 \\ (AG^{-1}A^T)^{-1}AG^{-1}\bar{r}_0 \end{pmatrix} = \begin{pmatrix} \bar{g}_0 \\ \underline{g}_0 \end{pmatrix},$$

where $\bar{g}_0$ is the projection of the vector $\bar{r}_0$ on the null space of A.
Since $\bar{p}_0 = \bar{g}_0$ belongs to the null space of A, we have that

$$Ax_1 = A(x_0 + \alpha\bar{p}_0) = A(x_0 + \alpha\bar{g}_0) = Ax_0 = b$$

and then

$$r_1 = \begin{pmatrix} \bar{r}_1 \\ 0 \end{pmatrix},$$

where $\bar{r}_1 = c - Hx_1 - A^T y_1$.
Let us proceed by induction and assume that $Ax_i = b$ and

$$p_i = \begin{pmatrix} \bar{p}_i \\ \underline{p}_i \end{pmatrix}$$

where $\bar{p}_i$ belongs to the null space of A. Then, since $x_{i+1} = x_i + \alpha \bar{p}_i$, we have that $Ax_{i+1} = b$ and

$$r_{i+1} = \begin{pmatrix} \bar{r}_{i+1} \\ 0 \end{pmatrix}. \tag{21}$$

Moreover, the preconditioned gradient g_{i+1} is given by

$$g_{i+1} = P^{-1} r_{i+1} = \begin{pmatrix} P_A \bar{r}_{i+1} \\ (AG^{-1}A^T)^{-1} AG^{-1} \bar{r}_{i+1} \end{pmatrix} = \begin{pmatrix} \bar{g}_{i+1} \\ \underline{g}_{i+1} \end{pmatrix},$$

where $\bar{g}_{i+1}$ belongs to the null space of A. Then we can conclude that $\bar{p}_{i+1}$ belongs to the null space of A, since, from the step (18) of the Algorithm 2.1, the direction p_{i+1} is computed as a linear combination of the preconditioned gradient g_{i+1} and the previous direction p_i. □

The previous theorem suggests that, if we choose an appropriate starting point, the use of the preconditioner P and the projection of the component $\bar{r}_i$ of the residual vector on the null space of A are two strictly related operations.

Remark: We observe that the component $A^T y_i$ of the residual $\bar{r}_i = c - Hx_i - A^T y_i$ lies in the range space of A^T, which is orthogonal to the null space of A; thus it does not affect, theoretically, the result of the projection $P_A \bar{r}_i$. The Example 1 in [12] shows that this fact could not be true from the numerical point of view: this fact will be further investigated in the next section.

Let us introduce the following notation: if Z is a $\bar{n} \times (\bar{n} - \bar{m})$ matrix whose columns form a basis for the null space of A, and A is a full row rank matrix, then every vector $x \in \mathbb{R}^{\bar{n}}$ admits a unique representation of the form

$$x = A^T x^\nu + Z x^\tau, \tag{22}$$

where $x^\nu = (AA^T)^{-1} Ax$ is the *normal component* of x and x^τ is the *tangenzial component* of x expressed in term of the basis Z. Indeed $\mathbb{R}^{\bar{n}}$ is the direct sum of the range space of A^T and of the null space of A.
If G is a symmetric nonsingular matrix, then we have also that

$$x^\nu = (AG^{-1}A^T)^{-1} AG^{-1} x,$$

since $G^{-1}Z$ is again a basis of the null space of A. Moreover we can also write

$$x = A^T x^\nu + GZx^{\tau_G}$$

For a convenient x^{τ_Z}, we have $GZx^{\tau_G} = GZ(Z^TGZ)^{-1}x^{\tau_Z}$, then $Z^Tx = x^{\tau_Z}$. Thus, we can write

$$x = A^Tx^\nu + GZ(Z^TGZ)^{-1}x^{\tau_Z} \tag{23}$$

and $Zx^\tau = GZ(Z^TGZ)^{-1}x^{\tau_Z}$.
We can obtain the tangential components with respect to the basis Z, GZ and $GZ(Z^TGZ)^{-1}$ by computing

$$\begin{aligned} x^\tau &= (Z^TZ)^{-1}Z^Tx \\ x^{\tau_G} &= (Z^TGZ)^{-1}Z^Tx \\ x^{\tau_Z} &= Z^Tx. \end{aligned}$$

The above relations are useful for the proof of the following theorem, where a different interpretation of Algorithm 2.1 is derived (see proof of Theorem 3.5 in [14]), under a weaker assumption: here we do not require that $b = 0$, $y_0 = 0$ and that Z is an orthonormal basis of the null space of A.

Theorem 2.2 *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. Let $x_* = A^Tx_*^\nu + Zx_*^\tau$ the first $\bar{n}$ components of the solution of the system (3). If we choose a starting point v_0 such that $Ax_0 = b$, then the tangential components $\{x_k^\tau\}$ of the elements of the sequence $\{x_k\}$ generated by Algorithm 2.1 applied to the system (3) are the elements of the sequence generated by the conjugate gradient method with the preconditioner Z^TGZ applied to the system*

$$Z^THZx^\tau = Z^Tc_z \tag{24}$$

where $c_z = c - HA^Tx_*^\nu$.

Proof. Since the starting point x_0 solves the system $Ax = b$, the previous theorem can be applied and the residuals r_i at each iterate i have the form (21). In particular, for the vector $\bar{r}_i$ we have

$$\bar{r}_i = (c - Hx_i) - A^Ty_i. \tag{25}$$

Since x_* is a solution of the system, we have that $Ax_* = b$ which implies $AA^Tx_*^\nu = b$. Moreover, we also have that $Ax_i = b$ for every index i, so that $AA^Tx_i^\nu = b$. Thus, we can conclude that $x_*^\nu = x_i^\nu$ for each i, because A is full row rank. By substituting the expression $x_i = A^Tx_i^\nu + Zx_i^\tau$ in (25), we obtain

$$\bar{r}_i = c - HZx_i^\tau - HA^Tx_i^\nu - A^Ty_i,$$

so that the tangential component of the residual $\bar{r}_i$ expressed with respect to the basis $GZ(Z^TGZ)^{-1}$is given by

$$\begin{array}{rcl} \bar{r}_i^{\tau z} & = & Z^T\bar{r}_i \\ & = & Z^T(c - HA^Tx_i^{\nu}) - Z^THZx_i^{\tau} \\ & = & Z^Tc_z - Z^THZx_i^{\tau} \end{array}$$

which is the residual of the system (24).
By exploiting the decomposition (23) of $\bar{r}^i$, we can write

$$r_i = \left(\begin{array}{c} GZ(Z^TGZ)^{-1}\bar{r}_i^{\tau z} + A^T\bar{r}_i^{\nu} \\ 0 \end{array} \right).$$

The projection operator P_A defined in (9), can be expressed by means of the null space basis Z as

$$P_Z = Z(Z^TGZ)^{-1}Z^T,$$

since, for every vector $u \in \mathbb{R}^{\bar{n}}$, using the formula (23), we have

$$P_Au = Z(Z^TGZ)^{-1}u^{\tau z} = P_Zu.$$

Exploiting the previous formulation of the projection operator and taking into account (20), it is easy to see that the preconditioned gradient computed at the step (14) has the following form:

$$g_i = P^{-1}r_i = \left(\begin{array}{c} Z(Z^TGZ)^{-1}\bar{r}_i^{\tau z} \\ \bar{r}_i^{\nu} \end{array} \right). \tag{26}$$

Thus, it follows that

$$r_i{}^Tg_i = \bar{r}_i^{\tau z^T}(Z^TGZ)^{-1}\bar{r}_i^{\tau z}. \tag{27}$$

Furthermore, recalling the previous theorem, the direction p_i can be written as

$$p_i = \left(\begin{array}{c} Z\bar{p}_i^{\tau} \\ \underline{p}_i \end{array} \right)$$

since $\bar{p}_i$ belongs to the null space of A; this implies that

$$Mp_i = \left(\begin{array}{c} HZ\bar{p}_i^{\tau} + A^T\underline{p}_i \\ 0 \end{array} \right)$$

from which we obtain

$$p_i{}^TMp_i = \bar{p}_i^{\tau T}(Z^THZ)\bar{p}_i^{\tau}. \tag{28}$$

The equalities (27) and (28) show that the coefficients α and β involved in the updating steps (12) and (18) of the Algorithm 2.1 are the same as in the CG algorithm with preconditioner $(Z^T GZ)$ applied to the system (24). Moreover, the tangential component of the vector $\bar{p}_{i+1} = Z\bar{p}_{i+1}^{\mathcal{T}}$ can be obtained as follows

$$\begin{array}{rcl} \bar{p}_i & = & Z\bar{p}_i^{\mathcal{T}} \\ & = & \bar{g}_i + \beta\bar{p}_{i-1} \\ & = & Z(Z^T GZ)^{-1}\bar{r}_i^{\mathcal{T}Z} + \beta Z\bar{p}_{i-1}^{\mathcal{T}} \end{array}$$

thus we can write

$$\bar{p}_i^{\mathcal{T}} = (Z^T GZ)^{-1}\bar{r}_i^{\mathcal{T}Z} + \beta\bar{p}_{i-1}^{\mathcal{T}}.$$

The last equality show that also the direction $\bar{p}_i^{\mathcal{T}}$ can be obtained by the updating rule of the CG algorithm with preconditioner $(Z^T GZ)$ applied to the system (24).
The previous remarks guarantee that the Algorithm 2.1 applied to the system (3) implicitly acts on the tangential components of the iterates as the CG method applied to the system (24) with preconditioner $(Z^T GZ)$. □

The previous result can be employed to derive an estimation of the absolute error and the finite termination property of the Algorithm 2.1. Indeed the Algorithm 2.1 finds the solution vector x_* after at most $\bar{n}-\bar{m}$ iterations, and for $1 \le i \le \bar{n} - \bar{m}$ the following estimation holds

$$\|x_i - x_*\| \le 2\sqrt{k}\left(\frac{1-\sqrt{k}}{1+\sqrt{k}}\right)^i \|x_0 - x_*\| \tag{29}$$

where $k = k(Z^T HZ(Z^T GZ)^{-1})$ and $\|\cdot\|$ is the euclidean norm.
Furthermore, under the assumptions of the previous theorem, the Algorithm 2.1 does not break down: indeed, the quantity δ computed at the step (10) actually has the form (27), and, if the matrix H is positive definite on the null space of A, δ is strictly positive.

Our implementation of the Algorithm 2.1 provides the direct factorization of the preconditioner P. We observe that this matrix can be factorized in a Cholesky–like form

$$L_{\bar{n}+\bar{m}} D L_{\bar{n}+\bar{m}}^T, \tag{30}$$

where $L_{\bar{n}+\bar{m}}$ is a lower triangular matrix with diagonal entries equal to one and D is a nonsingular diagonal matrix with $\bar{n}$ positive and $\bar{m}$ 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 symmetrically permuted matrix UPU^T can be factorized in the Cholesky–like form.

Nevertheless, we can obtain a factorization in the form (30) if we use for the matrix P the regularization technique described in [1]; in other words, instead of using the preconditioner P, we compute the factorization of

$$\bar{P} = P + \begin{pmatrix} R_1 & 0 \\ 0 & -R_2 \end{pmatrix}$$

where R_1 and R_2 are non negative diagonal matrices such that $U\bar{P}U^T$ admits a factorization of the form (30). The computation of R_1 and R_2 can be obtained during the factorization procedure. If a pivot d_i is too small ($|d_i| < 10^{-15} \max_{j<i} |d_j|$), we put $d_i = \sqrt{\epsilon}$ if $1 \leq i \leq \bar{n}$, or $d_i = -\sqrt{\epsilon}$ if $\bar{n} + 1 \leq i \leq \bar{n} + \bar{m}$, where ϵ is the machine precision.

The dynamic computation of the elements of R_1 and R_2 reduces the perturbation to a minimum. The Cholesky–like factorization of $\bar{P}$ can be obtained by a modification of the Ng and Peyton package. The modifications are described in [3]. 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.

2.2 Another projection algorithm

It is important to notice that the proof of Theorem 2.2 and the estimation (29) do not depend on the variable y, which represents the Lagrange multiplier of the equality constraints of the problem (2).

Indeed, the Algorithm 2.1 actually solves the problem

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

that we obtain by substituting (22) in (2). On the contrary of the methods proposed in [9], [13], [18], in this case we do not have to determine Z.

A similar approach is used in the Algorithm 2 in [12], where the PCG iteration is applied only to the primal variable x.

In this case, the residual vector is defined as $\tilde{r}^+ = Hx^i - c$, and, in general, in the algorithm, it 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 (3), 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, since $\tilde{g}_i = P_A \tilde{r}_i^+ = G^{-1}(\tilde{r}_i^+ - A^T v)$, where $v = \text{argmin}\|\tilde{r}_i^+ - A^T v\|_{G^{-1}}$, and, in finite arithmetic, $\tilde{r}_i^+ - A^T v$ might give rise to a significant cancellation of digits. This roundoff error leads the projected residual to do not belong exactly to the null space of A (see Example 1 in [12]).

In order to avoid this drawback, in [12] the authors propose a variant of the PCG algorithm, which, at each step, provides a least squares estimate of the normal component of the residual $A^T v_i$ and, then, it updates the residual by subtracting its normal 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 *Algorithm III (Preconditioned CG with residual update) in [12].*

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

$$\alpha \leftarrow \frac{\tilde{r}_i^T \tilde{g}_i}{\tilde{p}_i^T H \tilde{p}_i} \tag{31}$$

$$x_{i+1} = x_i + \alpha \tilde{p}_i$$

$$\tilde{r}_{i+1}^+ = \tilde{r}_i^+ + \alpha H \tilde{p}_i$$

$$v_{i+1} = \text{argmin}\|\tilde{r}_{i+1}^+ - A^T v\|_{G^{-1}} \tag{32}$$

$$\tilde{r}_{i+1} = \tilde{r}_{i+1}^+ - A^T v_{i+1} \tag{33}$$

$$\tilde{g}_{i+1} = P_A \tilde{r}_{i+1} \tag{34}$$

$$\beta \leftarrow \frac{\tilde{r}_{i+1}^T p_{i+1}}{\tilde{r}_i^T \tilde{g}_i}$$

$$\tilde{p}_{i+1} = \tilde{g}_{i+1} + \beta \tilde{p}_i$$

In order to compute the projection (34), we have to solve two system: first, we obtain v_{i+1} in (32) 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 (34), 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 $\tilde{r}^+ - A^T v$ belongs to the null space of A, so that the component u of the solution of the previous system is zero. In other words, in exact arithmetic, the desired projection of the residual $\tilde{r}$ is the vector $\tilde{r}$ itself; thus, the steps (32)–(33) can be considered as an iterative refinement step.

We observe that, at the iterate i of the Algorithm 2.2, we can obtain an estimate of the Lagrange multiplier as $y_i = -\sum_{k=0}^{i} v_k$ and, furthermore, the vector $\tilde{r}_i$ represents the residual of the first equation of the system (3): indeed we have $\tilde{r}_i = Hx_i + A^T y_i - c$. In exact arithmetic, the residual of the second equation of the system (3) should be the null vector, but operating in finite arithmetic, in general, this is not true.

In the Algorithm 2.1 the vector g^{i+1} is obtained by solving one system only

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

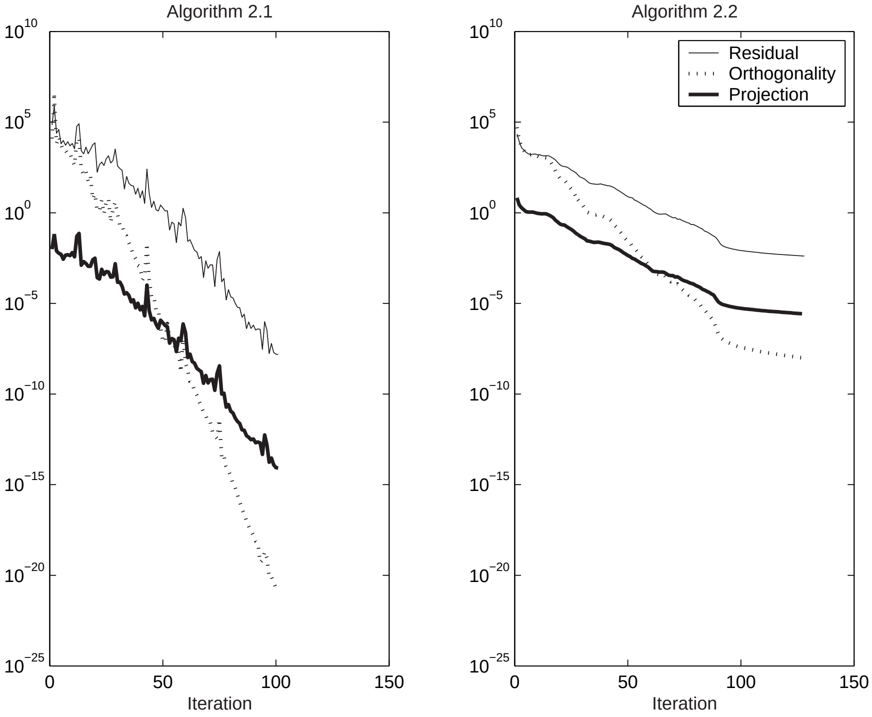
where $\bar{r}_{i+1} = c - Hx_{i+1} - A^T y_{i+1}$ and $\underline{r}_{i+1} = b - Ax_{i+1}$.

Also in this case, 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.

Thus, since the Algorithm 2.1 works with the primal and dual variables, it takes into account that $\bar{g}_{i+1}$ could not belong to the null space of A because of the roundoff errors in the component $\underline{r}_{i+1}$; on the other hand, the Algorithm 2.2 does not take into account of the dual variable, so that it partially controls the error on the projection of the residual vector $\tilde{r}^+_{i+1}$ with one step of iterative refinement. It is interesting to observe the effects of the finite precision on the Algorithms 2.1 and 2.2.

The figure 2.2 shows a comparison between the Algorithms 2.1 and 2.2 on the test problem CVXEQP3 of the CUTE collection [5] with $\bar{n} = 1000$ and $\bar{m} = 750$. For each iteration i, we have considered the following quantities: the norm of the residuals r_i and $\tilde{r}_i$, which indicate the progress towards the solution of the system(Residuals); the scalar products ${r_i}^T g_i$ and $\tilde{r}_i^T \tilde{g}_i$, which are measurements of the angle between the residual and g_i and $\tilde{g}_i$ respectively (Orthogonality); the quantities $\|A\bar{g}_i\|$ and $\|A\tilde{g}_i\|$, which tells us how precisely the projection is computed (Projection).

Figure 1: Test problem CVXEQP3



After 100 iterations, can observe that for the Algorithm 2.1, all the considered quantities are less than 10^{-8}, while for the Algorithm 2.2, the norm of the residual, is greater than 10^{-3}. Thus, the finite precision does not significantly influence the Algorithm 2.1, while it leads the Algorithm 2.2 to have less accuracy in the results.

3 Representations of the Newton system

In this section we describe four different formulations of the Newton system whose matrix is given in (6).

3.1 Full System

Following the approach in [21], [20], [6], by multiplying the last block of equations by S^{-1} and by performing a symmetric permutation on the second and fourth columns, we obtain a block system whose coefficient matrix is the following

$$\begin{pmatrix} Q & 0 & C^T & B^T \\ 0 & F^{-1} & I_m & 0 \\ C & I_m & 0 & 0 \\ B & 0 & 0 & 0 \end{pmatrix}, \tag{35}$$

where

$$F^{-1} = S^{-1}W. \tag{36}$$

We can consider the matrix (35) as a special case of (1) with

$$H = \begin{pmatrix} Q & 0 \\ 0 & F^{-1} \end{pmatrix} \text{ and } A = \begin{pmatrix} C & I_m \\ B & 0 \end{pmatrix}. \tag{37}$$

A sufficient condition for the nonsingularity of the matrix (35) is that

(A2) B is a full row rank matrix and $Q + C^T F^{-1} C$ is positive definite on the null space of B.

Let us prove that, under the hypothesis (A2), the matrix (35) is nonsingular. If we have

$$\begin{pmatrix} Q & 0 & C^T & B^T \\ 0 & F^{-1} & I_m & 0 \\ C & I_m & 0 & 0 \\ B & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ w \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

then, from the fourth equality, x belongs to the null space of B. From the third block of equations, we have $y = -Cx$ and, from the second block of equations, $w = -F^{-1}y + F^{-1}Cx$, so that

$$(Q + C^T F^{-1} C)x + B^T z = 0$$

If we consider

$$x^T(Q + C^T F^{-1} C)x + x^T B^T z = 0$$

since $Bx = 0$, from the hypothesis (A2), we have $x = 0$, $y = 0$, $w = 0$ and finally $B^T z = 0$. Since B is a full row rank matrix, $z = 0$. Then the matrix (34) is nonsingular. A sufficient condition so that the hypothesis (A2) is satisfied, is that the hypothesis (A1) holds.

3.2 Reduced system

From (35), by eliminating the second block of equations, we derive the second representation of the Newton system. In this case the coefficient matrix has the form

$$\begin{pmatrix} Q & C^T & B^T \\ C & -F & 0 \\ B & 0 & 0 \end{pmatrix} \tag{38}$$

The block structure of the matrix (38) is the same as in (1), with

$$H = \begin{pmatrix} Q & C^T \\ C & -F \end{pmatrix} \text{ and } A = (B \quad 0) \tag{39}$$

A special case of the system (38) can be obtained if we have $neq = 0$: this occurs when there are no equality constraints , or when the equality constraints in (4) are treated as range constraints, with upper and lower bounds that coincide, as in [7], [19].
In this case, the coefficient matrix is

$$M = H = \begin{pmatrix} Q & C^T \\ C & -F \end{pmatrix}. \tag{40}$$

A sufficient condition for the nonsingularity of the matrix (38) is that the hypothesis (A2) is satisfied. Indeed, if we have

$$\begin{pmatrix} Q & C^T & B^T \\ C & -F & 0 \\ B & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

then, from the last equality, x belongs to the null space of B.
From the second block of equations, we obtain $y = F^{-1}Cx$, so that, from the first block of equations, it follows that

$$(Q + C^T F^{-1} C)x + B^T z = 0.$$

Since x belongs to the null space of B, the previous equality yields $x^T(Q + C^T F^{-1} C)x = 0$ which, under the hypothesis (A2), implies $x = 0$ and $z = 0$. This proves the nonsingularity of the matrix (38).

The condition (A2) is consistent with the fact that the system

$$\begin{pmatrix} Q & C^T & B^T \\ C & -F & 0 \\ B & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} c \\ d \\ b \end{pmatrix} \tag{41}$$

represents the optimality conditions of the following equality constrained quadratic problem

$$\begin{array}{cl} \min & \frac{1}{2}x^TQx - c^Tx + \frac{1}{2}(Cx-d)^TF^{-1}(Cx-d) \\ Bx=b & \end{array} \tag{42}$$

Indeed, the optimality conditions for (42) are

$$\begin{array}{lcl} Qx + C^TF^{-1}(Cx-d) + B^Tz & = & c \\ Bx & = & b \end{array}$$

and by introducing a new variable y and the block of equations $y = F^{-1}(Cx-d)$, we obtain (41).

We point out that matrix in the reduced form has the block F, while in the full form it has the block F^{-1}.

3.3 Condensed form

By applying elimination techniques to the second block of equations in (38) we obtain the following coefficients matrix

$$\begin{pmatrix} \bar{Q} & B^T \\ B & 0 \end{pmatrix} \tag{43}$$

with $\bar{Q} = Q + C^TF^{-1}C$.
Also in this case, the hypothesis A2 is a sufficient condition for the nonsingularity of the matrix (43).

3.4 Active form

We consider also the approach followed in [15]; in this version of the interior–point method, a subdivision of the inequality constraints in two disjoint subsets is performed at each step, the active and the inactive constraints. The resulting system is obtained by eliminating only the equations related to the inactive constraints in the second block. Thus, the coefficient matrix has the following form:

$$H = \begin{pmatrix} \hat{Q} & C_a^T \\ C_a & -F_a \end{pmatrix} \text{ and } A = (B \quad 0) \tag{44}$$

where C_a is the jacobian matrix of the active inequality constraints, C_I indicates the jacobian matrix of the inactive inequality constraints and $\hat{Q} =$

$Q + C_I^T F_I^{-1} C_I$. Furthermore, F_a and F_I have the same meaning as in (36), but the slacks and the multipliers are only the ones related to the active and inactive constraints respectively.

In this case, the sizes of the blocks and the structure of the matrix can change at each iteration of the outer method.

3.5 The choice of the preconditioner

The Newton system can be solved with the PCG method described in the section 2. For the full and condensed representations the matrix P can be chosen as

$$\begin{pmatrix} D & 0 & C^T & B^T \\ 0 & F^{-1} & I_m & 0 \\ C & I_m & 0 & 0 \\ B & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \bar{D} & B^T \\ B & 0 \end{pmatrix} \tag{45}$$

where D and $\bar{D}$ are positive diagonal approximations of Q and $\bar{Q}$ respectively.

For the reduced and active form, the preconditioner can be chosen as

$$\begin{pmatrix} D & C^T & B^T \\ C & -F & 0 \\ B & 0 & 0 \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} \hat{D} & C_a^T & B^T \\ C_a & -F_a & 0 \\ B & 0 & 0 \end{pmatrix} \tag{46}$$

where $\hat{D}$ is a positive diagonal approximation of the matrix $\hat{Q}$.

We remember that $F = SW^{-1}$, where the elements of the diagonal matrices S and W are respectively the slack variables and the multipliers related to inequality constraints. We observe that the preconditioner in (46) related to the reduced form of the Newton system contains the matrix F while in the preconditioners in (45) related to the full and the condensed form of the Newton system, the inverse of F appears. The diagonal elements of F or F^{-1} affect the condition of the matrices M and P; indeed, in the last iterations of the outer optimization method, when we are near the solution, the diagonal elements of S corresponding to active inequality constraints and the ones of W corresponding to inactive inequality constraints assume gradually small values, very close to zero. If the method has a local superlinear convergence, this situation is not critical, since few outer steps produce the solution within the required accuracy [17]. Nevertheless the preconditioners in (45) can be more convenient when the problem has many inactive inequality constraints, while the preconditioner in (46) related to the reduced form can be more convenient in presence of many active constraints. The preconditioner in (46) related to the active form of the Newton system contains the diagonal entries of F related to the active inequality constraints and the inverse of diagonal entries of F corresponding to the inactive inequality constraints. This technique is used to assure that the entries of F_a and these of F_I^{-1} are bounded.

For the formulation (38), it is possible to prove some result analogous to the ones presented in the section 2.
Let us introduce the following notation: if $\bar{v}$ is a $\bar{n}$ vector, with $\bar{n} = n + m$, we will indicate its first n components as $\hat{v}$ and its last m components as $\check{v}$. Thus, if v is a $n + m + neq$ vector, we have

$$v = \begin{pmatrix} \bar{v} \\ \underline{v} \end{pmatrix} = \begin{pmatrix} \hat{v} \\ \check{v} \\ \underline{v} \end{pmatrix}.$$

Theorem 3.1 *Consider the PCG method applied to the system (41). Choosing the starting point such that $Bx_0 = b$ and $Cx_0 - Fy_0 = d$, if a breakdown does not occur, then the Algorithm 2.1 generates a sequence of points $\{x_i\}$ such that $Bx_i = b$ and $Cx_i - Fy_i = d$, so that*

$$r_i = \begin{pmatrix} \hat{r}_i \\ 0 \\ 0 \end{pmatrix}.$$

Moreover, at each iteration i, we have

$$\hat{r}_i = c + C^T F^{-1} d - (Q + C^T F^{-1} C)x_i - B^T z_i$$

and the component $\hat{p}_i$ of the direction p_i belongs to the null space of B; we have also that

$$\check{g}_i = F^{-1} C \hat{g}_i \qquad (47)$$

$$\check{p}_i = F^{-1} C \hat{p}_i \qquad (48)$$

We give only a sketch of the proof.
The first residual of the PCG procedure is a $n + m + neq$ vector given by

$$r_0 = \begin{pmatrix} \bar{r}_0 \\ 0 \end{pmatrix}$$

where

$$\bar{r}_0 = \begin{pmatrix} \hat{r}_0 \\ 0 \end{pmatrix},$$

and $\hat{r}_0 = c - Qx_0 - C^T y_0 - B^T z_0$.
Since y_0 solves the second equation of the system (41), then we have $y_0 = F^{-1}(Cx_0 - d)$, so that we can write

$$\hat{r}_0 = c + C^T F^{-1} d - (Q + C^T F^{-1} C)x_0 - B^T z_0.$$

The vector g_0 is computed as $g_0 = P^{-1} r_0$, thus $\bar{g}_0$ belongs to the null space of the matrix A defined in (39), but the null space of A is given by the vectors $(\hat{u}^T, \check{u}^T)^T$ such that $\hat{u}$ belongs to the null space of A and $\check{u} \in \mathbb{R}^m$. It follows that $\hat{g}$ belongs to the null space of B.
Moreover, the component $\check{g}_0$ solves the equation $C\hat{g}_0 - F\check{g}_0 = 0$, thus we have $\check{g}_0 = F^{-1}C\hat{g}_0$. Since $p_0 = g_0$, it follows that that $Bx_1 = B(x_0 + \alpha\hat{g}_0) = Bx_0 = b$, which implies $\underline{r}_1 = 0$.
Furthermore we have

$$\begin{array}{rcl} Cx_1 - Fy_1 & = & C(x_0 + \alpha\hat{g}_0) - F(y_0 + \alpha\check{g}_0) \\ & = & C(x_0 + \alpha\hat{g}_0) - F(y_0 + \alpha F^{-1}C\hat{g}_0) \\ & = & Cx_0 - Fy_0 \\ & = & d \end{array}$$

which yields $\check{r}_1 = 0$ and $\hat{r}_1 = c + C^T F^{-1} d - (Q + C^T F^{-1} C)x_1 - B^T z_1$.
Moreover we have $\check{p}_0 = F^{-1}C\hat{p}_0$.
If we assume that $Bx_i = b$ and $Cx_i - Fy_i = d$ so that $\check{r}_i = 0$ and $\underline{r}_i = 0$, it follows that the component $\check{g}_i$ of the vector $g_i = P^{-1} r_i$ satisfies the relation (47).
Moreover, assuming that $\check{p}_i = F^{-1}C\hat{p}_i$ and $\hat{p}_i$ belongs to the null space of B, we have that $Bx_{i+1} = B(x_i + \alpha\hat{p}_i) = b$, which implies $\underline{r}_{i+1} = 0$, and $Cx_{i+1} - Fy_{i+1} = C(x_i + \alpha\hat{p}_i) - F(y_i + \alpha F^{-1}C\hat{p}_i) = d$, which implies $\check{r}_i = 0$.
Since $g_{i+1} = P^{-1} r_{i+1}$, we have that $\hat{g}_{i+1} = P_A \hat{r}_{i+1}$ belongs to the null space of B, $\check{g}_{i+1} = F^{-1}C\hat{g}_{i+1}$. Then $\hat{p}_{i+1} = \hat{g}_{i+1} + \beta\hat{p}_i$ belongs to the null space of B.
Furthermore we have

$$\begin{array}{rcl} \check{p}_{i+1} & = & \check{g}_{i+1} + \beta\check{p}_i \\ & = & F^{-1}C\hat{g}_{i+1} + \beta F^{-1}C\hat{p}_i \\ & = & F^{-1}C(\hat{g}_{i+1} + \beta\hat{p}_i) \\ & = & F^{-1}C\hat{p}_{i+1} \end{array}$$

which yields (48).
□

If B is a full row rank matrix, Z is an $n\times(n-neq)$ matrix whose columns form a basis of the null space of B and $\bar{N}$ is a symmetric nonsingular matrix, we can write any n vector x as

$$\begin{aligned} x &= B^T x^\nu + Z x^\tau && (49)\\ x &= B^T x^\nu + \bar{N} Z x^{\tau_{\bar{N}}} && (50)\\ x &= B^T x^\nu + \bar{N} Z (Z^T \bar{N} Z)^{-1} x^{\tau_Z}. && (51) \end{aligned}$$

Given a symmetric nonsingular matrix D, for the normal component we can write

$$x^\nu = (BB^T)^{-1}Bx = (BD^{-1}B^T)^{-1}BD^{-1}x,$$

while the coefficients of the tangential components with respect to the different basis of the null space of B satisfy

$$\begin{aligned} x^\tau &= (Z^TZ)^{-1}Z^Tx\\ x^{\tau_{\bar{N}}} &= (Z^T\bar{N}Z)^{-1}Z^Tx\\ x^{\tau_Z} &= Z^Tx. \end{aligned}$$

Theorem 3.2 *Let Z a matrix whose columns form a basis for the null space of B, and assume that the hypothesis (A2) holds, so that the matrix $Z(Q + C^TF^{-1}C)Z^T$ is positive definite. Let $x_* = B^Tx_*^\nu + Zx_*^\tau$ the first n components of the solution of the system (41). Then, the tangential components $\{x_i^\tau\}$ of the sequence $\{x_i\}$ generated by the Algorithm 2.1 applied to the system (41) with a starting point such that $Bx_0 = b$ and $Cx_0 - Fy_0 = d$ are the elements of the sequence generated by the conjugate gradient method with the preconditioner $Z^T\bar{N}Z$, $\bar{N} = (D + C^TF^{-1}C)$, applied to the system*

$$Z^TNZx^\tau = Z^Tc_Z, \qquad (52)$$

where $N = Q + C^TF^{-1}C$ and $c_Z = (c + C^TF^{-1}d) - NB^Tx_*^\nu$ Thus, the conjugate gradient method finds the vector x_* after at most $n-neq$ iterations, and for $1 \le i \le \bar{n} - \bar{m}$ the following estimation holds

$$\|x_i - x_*\| \le 2\sqrt{k}\left(\frac{1-\sqrt{k}}{1+\sqrt{k}}\right)^i \|x_0 - x_*\| \qquad (53)$$

where $k = k(Z^TNZ(Z^T\bar{N}Z)^{-1})$.

Proof.
We can apply the previous result, so that

$$\hat{r}_i = (c + C^TF^{-1}d) - Nx_i - B^Tz_i,$$

from which, exploiting the formula (49) we obtain

$$\hat{r}_i = (c + C^T F^{-1} d) - NB^T x_i^\nu - NZx_i^\tau - B^T z_i.$$

Since $Bx_* = Bx_i = b$, it follows that $x_i^\nu = x_*^\nu$, thus the previous formula becomes

$$\hat{r}_i = c_Z - NZx_i^\tau - B^T z_i.$$

Its tangential component expressed in the basis $\bar{N}Z(Z^T\bar{N}Z)^{-1}$ can be obtained as follows:

$$\begin{array}{rcl} \hat{r}_i^{\tau z} & = & Z^T\hat{r}_i \\ & = & Z^T c_Z - Z^T\bar{N}Zx_i^\tau \end{array}$$

The inverse of the preconditioner P can be written as

$$P^{-1} = \left(\begin{array}{ccc} P_B & P_B C^T F^{-1} & D^{-1}B^T(BD^{-1}B^T)^{-1} \\ F^{-1}C^T P_B & F^{-1}CD^{-1}C^T F^{-1} & F^{-1}CD^{-1}B^T(BD^{-1}B^T)^{-1} \\ (BD^{-1}B^T)^{-1}BD^{-1} & (BD^{-1}B^T)^{-1}BD^{-1}C^T F^{-1} & -(BD^{-1}B^T)^{-1} \end{array} \right),$$

where P_B is the projection operator on the null space of B

$$P_B = \bar{N}^{-1} - \bar{N}^{-1}B^T(B\bar{N}^{-1}B^T)^{-1}B\bar{N}^{-1},$$

which can be expressed also as

$$P_Z = Z(Z^T\bar{N}Z)^{-1}Z^T.$$

If we write the residual in the form (51), we obtain

$$r_i = \left(\begin{array}{c} \hat{r}_i \\ 0 \\ 0 \end{array} \right) = \left(\begin{array}{c} B^T\hat{r}_i^\nu + \bar{N}Z(Z^T\bar{N}Z)^{-1}\hat{r}_i^{\tau z} \\ 0 \\ 0 \end{array} \right)$$

from which we can write

$$g_i = P^{-1}r_i = \left(\begin{array}{c} \hat{g}_i \\ \check{g}_i \\ \underline{g}_i \end{array} \right) = \left(\begin{array}{c} Z(Z^T\bar{N}Z)^{-1}\hat{r}_i^{\tau z} \\ F^{-1}C^T\hat{g}_i \\ \hat{r}_i^\nu \end{array} \right)$$

This implies that

$$r_i^T g_i = \hat{r}_i^{\tau z T}(Z^T\bar{N}Z)^{-1}\hat{r}_i^{\tau z}.$$

Furthermore, recalling that the component $\hat{p}_i$ belongs to the null space of B and taking into account of (48), we can write

$$p_i = \left(\begin{array}{c} Z\hat{p}_i^\tau \\ F^{-1}CZ\hat{p}_i^\tau \\ \underline{p}_i \end{array} \right),$$

from which we obtain

$$Mp_i = \begin{pmatrix} QZ\hat{p}_i^{\tau} + C^T F^{-1} C Z \hat{p}_i^{\tau} + B^T \underline{p}_i \\ 0 \\ 0 \end{pmatrix}$$

that implies

$$p_i^T M p_i = \hat{p}_i^{\tau T} (Z^T N Z) \hat{p}_i^{\tau}.$$

Finally, we observe that the vector $\hat{p}_i^{\tau}$ can be written as

$$\hat{p}_i^{\tau} = (Z^T \bar{N} Z)^{-1} \hat{r}_i^{\tau z} + \beta \hat{p}_{i-1}^{\tau}$$

which is the updating step of the PCG method with preconditioner $(Z^T\bar{N}Z)$ applied to the system (52). This implies that the Algorithm 2.1 applied to the system (41) is the PCG method with preconditioner $(Z^T\bar{N}Z)$ applied to the system (52); thus the estimate (53) holds. □

Finally, for the special case (40), the Algorithm 2.1 with the following preconditioner

$$\begin{pmatrix} D & C^T \\ C & -F \end{pmatrix}, \tag{54}$$

applied to the system

$$\begin{pmatrix} Q & C^T \\ C & -F \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c \\ d \end{pmatrix},$$

is equivalent to the application of the PCG method to the system

$$(Q + C^T F^{-1} C)x = c + C^T F^{-1} d$$

with preconditioner $D + C^T F^{-1} C$ (Theorem 3.1 in [7]). The system is nonsingular under the hypothesis (A2)

4 Numerical results

The aim of our numerical experience is to compare the effectiveness of the PCG method as solver for the Newton system formulated in the four cases described in section 3. In particular, we are interested on how much the representation of the Newton system can influence the performance of the method. Furthermore, we use the PCG method as inner solver in an interior–point method: we consider the inexact Newton interior–point algorithm described in [2], which at each step has to solve the Newton system associated

to the KKT conditions of the NLP problem (4).
The numerical results presented in the following have been carried out by coding this algorithm 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.

In order to obtain a significant comparison, we build a set of NLP problems, with a large number of inequality constraints, listed in table 1. The columns nl and nu report the number of lower and upper bounds on the components of the variable x. Except for the problem Svanberg, which belongs to the CUTE collection, the other test problems have been obtained by modifying the nonlinearly equality constrained problems described in [16]: some or all of the original equality constraints have been changed in inequality constraints.

The first comparison in table 2 shows the performances of the PCG method on the solution of the Newton system arising at the last iterate of the interior–point code. The starting point is the same in all cases. The stopping criterion for the PCG procedure is

$$\|r_i\| \leq 10^{-12}.$$

The system has been formulated in the four different representations described in the previous section: “3x3” denotes the reduced form, “2x2” is the condensed system, “4x4” is the full system while “Luk” indicates the active approach.
In this case, we say that the i^{th} inequality constraint is inactive if the following condition is satisfied:

$$w_i \leq 10^{-5} s_i.$$

The first two columns of the table 2 indicates the number of the active (ma) and inactive (mi) inequality constraint, thus the dimension of the system in the active form is given by $n + ma + neq$.
The table reports the comparison in terms of iterations number and execution time (in seconds), while the symbol “ * ” indicates that the tolerance of 10^{-12} was not satisfied after $\bar{n}$ iterations.
The table 3 summarizes the results of the comparison, reporting the number of test problems in which the Newton system in the different representations has been solved by the PCG method obtaining the best and worst performance. The representations of the Newton system which gives the best result are the reduced and the active form, while the less valid approach

seems to be the reduction in condensed form. An explanation of this fact could be that the preconditioner of the system in condensed form is obtained by approximating the matrix $\hat{Q}$, which in general is not sparse, with a diagonal matrix, so that the approximation could be very poor.
Furthermore, we can notice that the best performance obtained with the reduction in condensed form have been obtained in the test problems with many inactive constraints, for example the problem Lukvli12. This could be explained by observing that in this case the diagonal matrix F^{-1} has small elements, so that the dense part of $\hat{Q} = Q + C^T F^{-1} C$ is in some way "weighted" by small quantities.

The table 4 contains the time comparison of four different version of the interior–point method described in [2] obtained by representing the Newton system in the forms presented in the section 3.
In this case, the termination rule 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 5 we report the summary of this experimentation: the interior–point algorithm performs better if the Newton system is represented in reduced form or in active form. The behaviour of the PCG method for these forms is very similar and we observe that, generally, the PCG method is convenient for test problems with many active inequality constraints. Also the PCG method applied to the full system is convenient but, in general, has a worse behaviour than that of the reduced and the active form. For the condensed form of the Newton system, the PCG method is convenient when there are many inactive inequality constraints (Lukvlie17, Lukvlie18, Lukvli10) ($\bar{Q}$ is reduced to Q at the last iterations), but, in the opposite case, we observe worse performance or failure.

Table 1: Test Problems

N	TEST	n	neq	m	nl	nu
1	Lukvlie3	50000	1	1	0	0
2	Lukvlie4	50000	24999	24999	0	0
3	Lukvlie6	50001	12500	12500	0	0
4	Lukvlie7	50000	2	2	0	0
5	Lukvlie10	50000	24999	24999	0	0
6	Lukvlie11	49997	16665	16665	0	0
7	Lukvlie14	49997	16665	16665	0	0
8	Lukvlie16	49997	12499	24998	0	0
9	Lukvlie17	49997	12499	24998	0	0
10	Lukvlie18	49997	12499	24998	0	0
11	Lukvli2	50000	0	49993	0	0
12	Lukvli3	50000	0	2	0	0
13	Lukvli4	50000	0	49998	0	0
14	Lukvli6	49999	0	24999	0	0
15	Lukvli7	50000	0	4	0	0
16	Lukvli10	50000	0	49998	0	0
17	Lukvli11	49997	0	33330	0	0
18	Lukvli12	49997	0	37497	0	0
19	Lukvli13	49997	0	33330	0	0
20	Lukvli14	49997	0	33330	0	0
21	Lukvli15	49997	0	37497	0	0
22	Lukvli16	49997	0	37497	0	0
23	Lukvli18	49997	0	37497	0	0
24	Svanberg	50000	0	50000	50000	50000

Table 2: Numerical solution of one Newton system

			CG iterations				Execution time			
ma	mi	Prob	3x3	2x2	4x4	Luk	3x3	2x2	4x4	Luk
1	0	Lukvlie3	6	25	6	6	0.1	0.6	0.1	0.1
12498	12501	Lukvlie4	14	233	15	14	0.8	9.4	0.9	0.8
12500	0	Lukvlie6	9	203	9	9	0.6	11.5	0.6	0.6
2	0	Lukvlie7	4	13	10	4	0.1	0.2	0.1	0.1
24999	0	Lukvlie10	3	106	3	3	0.2	4.0	0.2	0.2
16664	1	Lukvlie11	10	15	22	11	0.5	0.5	1.0	0.5
16665	0	Lukvlie14	2	*	2	2	0.1	*	0.1	0.1
24997	1	Lukvlie16	10	21	14	9	0.4	0.6	0.6	0.4
1	24997	Lukvlie17	10	48	10	10	0.4	1.4	0.4	0.3
0	24998	Lukvlie18	3	5	3867	3	0.1	0.2	154.0	0.1
25000	24993	Lukvli2	12	1266	305	13	1.2	78.6	26.8	1.4
1	1	Lukvli3	7	10	7	7	0.2	0.2	0.2	0.2
24998	2500	Lukvli4	16	171	17	16	1	6.1	1	1
24999	0	Lukvli6	12	255	12	12	0.8	13.8	0.9	0.7
3	1	Lukvli7	4	27	7	4	0.1	0.3	0.1	0.1
0	33330	Lukvli10	23	24	34	25	1.0	0.7	1.6	0.8
0	37497	Lukvli11	*	*	*	*	*	*	*	*
0	33330	Lukvli12	11	11	12	11	0.5	0.4	0.6	0.4
33329	1	Lukvli13	4	960	5	4	0.2	25,0	0.3	0.1
37492	5	Lukvli14	36	68	46	37	1.7	2.2	2.4	1.9
37496	1	Lukvli15	11	21	26	11	0.5	0.6	1.2	0.5
24998	12499	Lukvli16	5	10	*	5	0.2	0.3	*	0.2
49998	3	Lukvli18	3	109	3	3	0.2	3.6	0.2	0.2
40387	9613	Svanberg	1	*	2	1	0.2	*	0.3	0.2

Table 3: Summary

	CG iterations				Execution time			
	3x3	2x2	4x4	Luk	3x3	2x2	4x4	Luk
Best	22	1	8	19	18	4	9	20
Worst	0	16	5	0	0	14	8	0
Failures	1	3	2	1				

Table 4: Execution time in seconds for an interior–point algorithm

N	Problem	3x3	2x2	4x4	Luk
1	Lukvlei3	11.5	11.1	11.1	11.5
2	Lukvlei4	51.7	*	57.7	51.7
3	Lukvlei6	161.1	534	612	161.1
4	Lukvlei7	6.6	8.0	10.5	6.6
5	Lukvlei10	*	46.4	*	*
6	Lukvlei11	24.3	27.0	26.6	25.5
7	Lukvlei14	*	*	482.7	*
8	Lukvlei16	21.8	24.1	22.1	20.8
9	Lukvlei17	46.3	16.4	*	167.9
10	Lukvlei18	12.1	11.7	12.7	12.1
11	Lukvli2	127.3	*	83.1	126.8
12	Lukvli3	8.6	14.4	9.3	8.7
13	Lukvli4	54.3	1408.4	51.9	54.9
14	Lukvli6	177.7	564.0	132.5	177.0
15	Lukvli7	9.0	11.9	10.8	9.1
16	Lukvli10	*	41.2	*	*
17	Lukvli11	211.1	198.9	615.0	189.5
18	Lukvli12	55.54	170.5	43.6	50.9
19	Lukvli13	310.1	349.2	22.0	319.7
20	Lukvli14	20.2	*	22.0	20.4
21	Lukvli15	187.4	*	102.5	205.5
22	Lukvli16	20.0	20.7	22.6	19.9
23	Lukvli18	12.2	13.1	13.4	13.5
24	Svanberg	59.0	*	60.2	63.1

References

[1] A. Altman and J. Gondzio, Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization, Optim. Methods Softw., vol. 11–12, 1999, 275–302.

[2] S. Bonettini, E. Galligani and V. Ruggiero, Inner solvers for interior point methods for large scale nonlinear programming, Technical report n. 64, Dipartimento di Matematica Pura ed Applicata, Università di Modena e Reggio Emilia, 2005. To appear on *Computational Optimization and Applizations.*

Table 5: Summary

	2x2	3x3	4x4	Luk
Best	5	9	7	6
Worst	8	2	7	5
Failure	6	3	3	3

[3] S. Bonettini and V. Ruggiero, Some iterative methods for the solution of a reduced symmetric indefinite KKT system, Technical report n. 60, Dipartimento di Matematica Pura ed Applicata, Università di Modena e Reggio Emilia, 2005. To appear on *Computational Optimization and Applizations.*

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

[5] I. Bongartz, A.R. Conn, N. Gould and Ph. L. Toint (1995). *CUTE: Constrained and Unconstrained Testing Environnment*, ACM Transactions on Mathematical Software, **21**, 123–160.

[6] R. H. Byrd, J. C. Gilbert and J. Nocedal, A trust region method based on interior point techniques for nonlinear programming, *Mathematical Programming*, **89**, 2000, 149–185.

[7] S. Cafieri, M. D'Apuzzo,V. De Simone and D. di Serafino, (2004) On the iterative solution of KKT systems in potential reduction software for large scale quadratic problems to appear on Computational Optimization and Applications.

[8] E.D. Dolan, J. J. Moré and T. S. Munson (2004). *Benchmarking optimization software with COPS 3.0*, Technical Report ANL/MCS-TM-273, Argonne National Laboratory, Illinois, USA.

[9] J. Dunn, (1993). Second–order multiplier calculations for optimal control problems and related large scale nonlinear programs, SIAM Journal on Optimization, **3**, 489–502.

[10] 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.

[11] Durazzi C., Ruggiero V.; *Indefinitely preconditioned conjugate gradient method for large sparse equality and inequality constrained quadratic problems*, Numer. Linear Algebra Appl., 10 (2003), 673-688 .

[12] 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.*, vol. 23, 2001, 1376–1395.

[13] D. James (1992). Implicit nullspace iterative methods for constrained least square problems, SIAM Journal on Matrix Analysis and Applications, **13**, 962–978.

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

[15] L. Lukšan, C. Matonoha and J. Vlček, Interior–point method for nonlinear nonconvex optimization, *Numerical Linear Algebra with Applications*, vol. 11, 5-6, 2004, 431–453.

[16] L. Lukšan and J. Vlček (1999). Sparse and partially separable test problems for unconstrained and equality constrained optimization, Technical report 767, Institute of Computer Science, Academy of Science of the Czech Republic.

[17] F.A. Potra F.A. and S.J. Wright, Interior–point methods, Journal of Computational and Applied Mathematics., 124 (2000) 281–302.

[18] Ph. L. Toint and D. Tuyttens (1990). On large–scale nonlinear network optimization, Mathematical Programming, Series B, **48**, 125–159.

[19] R. J. Vanderbei and D.F. Shanno, An interior-point algorithm for nonconvex nonlinear programming, *Computational Optimization and Applications*, vol. 13, 1999, 231–252.

[20] A. Wächter and L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Technical Report RC 23149, IBM T.J. Watson Research Center, Yorktown Heights, USA, 2004. To appear in *Mathematical Programming.*

[21] 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, Report OTC 6/2003, Optimization Technology Center, Northwestern University, Evanston, IL 60208, USA, 2003. To appear in *Mathematical Programming, Series A*.