**RAL-TR-2004-037**

# On iterative methods and implicit-factorization preconditioners for regularized saddle-point systems

H. Sue Dollar[1], Nicholas I. M. Gould[2,3],
Wil H. A. Schilders[4] and Andrew J. Wathen[1]

**ABSTRACT**

We consider conjugate-gradient like methods for solving block symmetric indefinite linear systems that arise from saddle point problems or, in particular, regularizations thereof. Such methods require preconditioners that preserve certain sub-blocks from the original systems but allow considerable flexibility for the remaining blocks. We construct fourteen families of implicit factorizations that are capable of reproducing the required sub-blocks and (some) of the remainder. These generalize known implicit factorizations for the unregularized case. Improved eigenvalue clustering is possible if additionally some of the non-crucial blocks are reproduced. Numerical experiments confirm that these implicit-factorization preconditioners can be very effective in practice.

---

[1] Oxford University Computing Laboratory, Numerical Analysis Group, Wolfson Building, Parks Road, Oxford, OX1 3QD, England, UK. Email: Sue.Dollar@comlab.ox.ac.uk & Andy.Wathen@comlab.ox.ac.uk . Current reports available from "web.comlab.ox.ac.uk/oucl/publications/natr/index.html".

[2] Computational Science and Engineering Department, Rutherford Appleton Laboratory, Chilton, Oxfordshire, OX11 0QX, England, UK. Email: n.i.m.gould@rl.ac.uk . Current reports available from "www.numerical.rl.ac.uk/reports/reports.shtml".

[3] This work was supported by the EPSRC grants GR/S42170.

[4] Philips Research Laboratories; Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands. Email: wil.schilders@philips.com . Also, Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, PO Box 513, 5600 MB Eindhoven, The Netherlands.

Computational Science and Engineering Department
Atlas Centre
Rutherford Appleton Laboratory
Oxfordshire OX11 0QX

June 13, 2005

# 1 Introduction

Given a symmetric $n$ by $n$ matrix $H$, a symmetric $m$ by $m$ ($m \leq n$) matrix $C$ and a full-rank $m$ ($\leq n$) by $n$ matrix $A$, we are interested in solving structured linear systems of equations

$$\begin{pmatrix} H & A^T \\ A & -C \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = - \begin{pmatrix} g \\ 0 \end{pmatrix}, \tag{1.1}$$

by iterative methods, in which preconditioners of the form

$$M_G = \begin{pmatrix} G & A^T \\ A & -C \end{pmatrix} \tag{1.2}$$

are used to accelerate the iteration for some suitable symmetric $G$. There is little loss of generality in assuming the right-hand side of (1.1) has the form given rather than with the more general

$$\begin{pmatrix} H & A^T \\ A & -C \end{pmatrix} \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix}. \tag{1.3}$$

For, so long as we have some mechanism for finding an initial $(x_0, y_0)$ for which $Ax_0 - Cy_0 = c$, linearity of (1.1) implies that $(\bar{x}, \bar{y}) = (x_0 - x, y_0 - y)$ solves (1.3) when $b = g + Hx_0 + A^T y_0$. In particular, since we intend to use the preconditioner (1.2), solving

$$\begin{pmatrix} G & A^T \\ A & -C \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} 0 \\ c \end{pmatrix} \quad \text{or} \quad = \begin{pmatrix} b \\ c \end{pmatrix}$$

to find suitable $(x_0, y_0)$ are distinct possibilities.

When $C = 0$, (1.2) is commonly known as a constraint preconditioner [33] and in this case systems of the form (1.1) arise as stationarity (KKT) conditions for equality-constrained optimization [37, §18.1], in mixed finite-element approximation of elliptic problems [5], including in particular problems of elasticity [38] and incompressible flow [22], as well as other areas. In practice $C$ is often positive semi-definite (and frequently diagonal)—such systems frequently arise in interior-point and regularization methods in optimization, the simulation of electronic circuits [43] and other related areas. Although such problems may involve $m$ by $n$ $A$ with $m > n$, this is not a restriction for in this case we might equally solve

$$\begin{pmatrix} C & A \\ A^T & -H \end{pmatrix} \begin{pmatrix} y \\ -x \end{pmatrix} = \begin{pmatrix} -c \\ b \end{pmatrix},$$

for which $A^T$ has more columns than rows. We place no restrictions on $H$, although we recognise that in some applications $H$ may be positive (semi-) definite.

## Notation

Let $I$ by the (appropriately-dimensioned) identity matrix. Given a symmetric matrix $M$ with, respectively, $m_+$, $m_-$ and $m_0$ positive, negative and zero eigenvalues, we denote its inertia by $\text{In}(M) = (m_+, m_-, m_0)$.

# 2 Suitable iterative methods

While it would be perfectly possible to apply a preconditioned iterative method like MINRES or QMR to (1.1) with the preconditioner (1.2), in many cases—including those mentioned above—it is actually possible to use the more efficient and effective preconditioned conjugate-gradient (PCG) method instead. We shall focus on this approach in this paper, and thus need to derive conditions for which PCG is an appropriate method.

Suppose that $C$ is of rank $l$, and that we find a decomposition

$$C = EDE^T, \tag{2.1}$$

where $E$ is $m$ by $l$ and $D$ is $l$ by $l$ and invertible—either a spectral decomposition or an $LDL^T$ factorization with pivoting are suitable, but the exact form is not relevant. In this case, on defining additional variables

$$z = -DE^Ty,$$

we may rewrite (1.1) as

$$\begin{pmatrix} H & 0 & A^T \\ 0 & D^{-1} & E^T \\ A & E & 0 \end{pmatrix} \begin{pmatrix} x \\ z \\ y \end{pmatrix} = \begin{pmatrix} g \\ 0 \\ 0 \end{pmatrix}. \tag{2.2}$$

Noting the trailing zero block in the coefficient matrix of (2.2), we see that the required $(x, z)$ components of the solution lie in the null-space of $(A \;\; E)$.

Let the columns of the matrix

$$N = \begin{pmatrix} N_1 \\ N_2 \end{pmatrix}$$

form a basis for this null space. Then

$$\begin{pmatrix} x \\ z \end{pmatrix} = \begin{pmatrix} N_1 \\ N_2 \end{pmatrix} w \tag{2.3}$$

for some $w$, and (2.2) implies

$$H_N w = N_1^T g. \tag{2.4}$$

where

$$H_N \stackrel{\text{def}}{=} N_1^T H N_1 + N_2^T D^{-1} N_2. \tag{2.5}$$

Since we would like to apply PCG to solve (2.4), our fundamental assumption is then that

**A1** the matrix $H_N$ is positive definite.

Fortunately assumption **A1** is often easy to verify. For we have

**Theorem 2.1.** *Suppose that the coefficient matrix $M_H$ of (1.1) is non-singular and has $m_{H-}$ negative eigenvalues and that $C$ has $c_-$ negative ones, then* **A1** *holds if and only if*

$$m_{H-} + c_- = m. \tag{2.6}$$

**Proof.** It is well known [28, Thm. 2.1] that under assumption **A1** the coefficient matrix $E_H$ of (2.2) has inertia $(n + l, m, 0)$. The result then follows directly from Sylvester's law of inertia, since then $\text{In}(E_H) = \text{In}(D^{-1}) + \text{In}(M_H)$ and $D^{-1}$ has as many negative eigenvalues as $C$ has. □

Under assumption A1, we may apply the PCG method to find $w$, and hence recover $(x, z)$ from (2.3). Notice that such an approach does not determine $y$, and additional calculations may need to be performed to recover it if it is required.

More importantly, it has been shown [8, 11, 30, 40] that rather than computing the iterates explicitly within the null-space via (2.3), it is possible to perform the iteration in the original $(x, z)$ space so long as the preconditioner is chosen carefully. Specifically, let $G$ be any symmetric matrix for which

**A2** the matrix

$$G_N \stackrel{\text{def}}{=} N_1^T G N_1 + N_2^T D^{-1} N_2 \tag{2.7}$$

is positive definite,

which we can check using Theorem 2.1. Then the appropriate projected preconditioned conjugate-gradient (PPCG) algorithm is as follows [30]:

---

**Projected Preconditioned Conjugate Gradients (variant 1):**

Given $x = 0$, $z = 0$ and $h = 0$, solve

$$\begin{pmatrix} G & 0 & A^T \\ 0 & D^{-1} & E^T \\ A & E & 0 \end{pmatrix} \begin{pmatrix} r \\ d \\ u \end{pmatrix} = \begin{pmatrix} g \\ h \\ 0 \end{pmatrix}, \tag{2.8}$$

and set $(p, v) = -(r, d)$ and $\sigma = g^T r + h^T d$.
Iterate until convergence:

$$
\begin{array}{llcl}
\text{Form} & Hp & \text{and} & D^{-1}v \\
\text{Set} & \alpha & = & \sigma/(p^T H p + v^T D^{-1} v). \\
\text{Update} & x & \leftarrow & x + \alpha p, \\
 & z & \leftarrow & z + \alpha v, \\
 & g & \leftarrow & g + \alpha H p \\
\text{and} & h & \leftarrow & h + \alpha D^{-1} v.
\end{array}
$$

$$
\text{Solve} \begin{pmatrix} G & 0 & A^T \\ 0 & D^{-1} & E^T \\ A & E & 0 \end{pmatrix} \begin{pmatrix} r \\ d \\ u \end{pmatrix} = \begin{pmatrix} g \\ h \\ 0 \end{pmatrix}.
$$

$$
\begin{array}{llcl}
\text{Set} & \sigma_{\text{new}} & = & g^T r + h^T d \\
\text{and} & \beta & = & \sigma_{\text{new}}/\sigma. \\
\text{Update} & \sigma & \leftarrow & \sigma_{\text{new}}, \\
 & p & \leftarrow & -r + \beta p \\
\text{and} & v & \leftarrow & -d + \beta v.
\end{array}
$$

---

The scalar $\sigma$ gives an appropriate optimality measure [30], and a realistic termination rule is to stop when $\sigma$ is small relative to its original value.

While this method is acceptable when a decomposition (2.1) of $C$ is known, it is preferable to be able to work directly with $C$. To this end, suppose that at each iteration

$$
h = -E^T a, \quad v = -DE^T q \text{ and } d = -DE^T t
$$

for unknown vectors $a$, $q$ and $t$—this is clearly the case at the start of the algorithm. Then, letting $w = Ca$, it is straightforward to show that $t = u + a$, and that we can replace our previous algorithm with the following equivalent one:

---

**Projected Preconditioned Conjugate Gradients (variant 2):**

Given $x = 0$, and $a = w = 0$, solve

$$
\begin{pmatrix} G & A^T \\ A & -C \end{pmatrix} \begin{pmatrix} r \\ u \end{pmatrix} = \begin{pmatrix} g \\ w \end{pmatrix},
$$

and set $p = -r$, $q = -u$ and $\sigma = g^T r$.
Iterate until convergence:

$$\begin{array}{llcl}
\text{Form} & Hp \text{ and } Cq & & \\
\text{Set} & \alpha & = & \sigma/(p^T Hp + q^T Cq). \\
\text{Update} & x & \leftarrow & x + \alpha p, \\
 & a & \leftarrow & a + \alpha q, \\
 & g & \leftarrow & g + \alpha Hp \\
\text{and} & w & \leftarrow & w + \alpha Cq.
\end{array}$$

$$\text{Solve} \quad \begin{pmatrix} G & A^T \\ A & -C \end{pmatrix} \begin{pmatrix} r \\ u \end{pmatrix} = \begin{pmatrix} g \\ w \end{pmatrix}.$$

$$\begin{array}{llcl}
\text{Set} & t & = & a + u \\
 & \sigma_{\text{new}} & = & g^T r + t^T w \\
 & \beta & = & \sigma_{\text{new}}/\sigma. \\
\text{Update} & \sigma & \leftarrow & \sigma_{\text{new}}, \\
 & p & \leftarrow & -r + \beta p \\
\text{and} & q & \leftarrow & -t + \beta q.
\end{array}$$

---

Notice now that $z$ no longer appears, and that the preconditioning is carried out using the matrix $M_G$ mentioned in the introduction. Also note that although this variant involves two more vectors than its predecessor, $t$ is simply used as temporary storage and may be omitted if necessary, while $w$ may also be replaced by $Ca$ if storage is tight.

When $C = 0$, this is essentially the algorithm given by [30], but for this case the updates for $v$ and $w$ are unnecessary and may be discarded. At the other extreme, when $C$ is non singular the algorithm is precisely that proposed by [29, Alg. 2.3], and is equivalent to applying PCG to the system

$$(H + A^T C^{-1} A)x = g$$

using a preconditioner of the form $G + A^T C^{-1} A$.

Which of the two variants is prefereable depends on whether we have a decomposition (2.1) and whether $l$ is small relative to $m$: the vectors $h$ and $v$ in the first variant are of length $l$, while the corresponding $a$ and $q$ in the second are of length $m$. Notice also that although the preconditioning steps in the first variant require that we solve

$$\begin{pmatrix} G & 0 & A^T \\ 0 & D^{-1} & E^T \\ A & E & 0 \end{pmatrix} \begin{pmatrix} r \\ d \\ u \end{pmatrix} = \begin{pmatrix} g \\ h \\ 0 \end{pmatrix}, \tag{2.9}$$

this is entirely equivalent to solving

$$\begin{pmatrix} G & A^T \\ A & -C \end{pmatrix} \begin{pmatrix} r \\ u \end{pmatrix} = \begin{pmatrix} g \\ w \end{pmatrix},$$

where $w = -EDh$, and recovering

$$d = D(h - E^T v).$$

Thus our remaining task is to consider how to build suitable and effective preconditioners of the form (1.2). We recall that it is the distribution of the generalized eigenvalues $\lambda$ for which

$$H_N \bar{v} = \lambda G_N \bar{v} \tag{2.10}$$

that determines the convergence of the preceding PPCG algorithms, and thus we will be particularly interested in preconditioners which cluster these eigenvalues. In particular, if we can efficiently compute $G_N$ so that there are few distinct eigenvalues $\lambda$ in (2.10), then PPCG convergence (termination) will be rapid.

# 3 Eigenvalue considerations

We first consider the spectral implications of preconditioning (1.1) by (1.2).

**Theorem 3.1.** [16, Thm. 3.1] or, in special circumstances, [3, 41]. *Suppose that $M_H$ is the coefficient matrix of (1.1). Then $M_G^{-1} M_H$ has $m$ unit eigenvalues, and the remaining $n$ eigenvalues satisfy*

$$(H - \lambda G)v = (\lambda - 1)A^T w \quad \textit{where} \quad Av - Cw = 0.$$

*If $C$ is invertible, the non-unit eigenvalues satisfy*

$$(H + A^T C^{-1} A)v = \lambda (G + A^T C^{-1} A)v. \tag{3.1}$$

In order to improve upon this result, we first consider the special case in which $C = 0$.

## 3.1 The case $C = 0$

In the extreme case where $C = 0$, we have previously obtained [17] a number of significantly better results, which we now summarise. Suppose that

$$K_H = \begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix} \text{ and } K_G = \begin{pmatrix} G & A^T \\ A & 0 \end{pmatrix}.$$

The requirement **A2** and Theorem 2.1 imply that

$$\text{In}(K_G) = (n, m, 0). \tag{3.2}$$

This leads to

**Theorem 3.2.** [33, Thm. 2.1] or, for diagonal $G$, [34, Thm. 3.3]. *Suppose that $N$ is any ($n$ by $n-m$) basis matrix for the null-space of $A$. Then $K_G^{-1}K_H$ has $2m$ unit eigenvalues, and the remaining $n-m$ eigenvalues are those of the generalized eigenproblem*

$$N^T H N v = \lambda N^T G N v. \tag{3.3}$$

The eigenvalues of (3.3) are real since (3.2) is equivalent to $N^T GN$ being positive definite [7, 28].

Although we are not expecting or requiring that $G$ (or $H$) be positive definite, it is well-known that this is often not a significant handicap.

**Theorem 3.3.** [1, Cor. 12.9, or 13, for example]. *The inertial requirement (3.2) holds for a given $G$ if and only if there exists a positive semi-definite matrix $\bar{\Delta}$ such that $G + A^T \Delta A$ is positive definite for all $\Delta$ for which $\Delta - \bar{\Delta}$ is positive semi-definite.*

Since any preconditioning system

$$\begin{pmatrix} G & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} r \\ s \end{pmatrix} \tag{3.4}$$

may equivalently be written as

$$\begin{pmatrix} G + A^T \Delta A & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} u \\ w \end{pmatrix} = \begin{pmatrix} r \\ s \end{pmatrix} \tag{3.5}$$

where $w = v - \Delta A u$, there is little to be lost (save sparsity in $G$) in using (3.5), with its positive-definite leading block, rather than (3.4) [17, 27, 32, 34]. Notice that perturbations of the form $G + A^T \Delta A$ do not change the eigenvalue distribution alluded to in Theorem 3.2, since if $H(\Delta_H) = H + A^T \Delta_H A$ and $G(\Delta_G) = G + A^T \Delta_G A$, for (possibly different) $\Delta_H$ and $\Delta_G$,

$$N^T H(\Delta_H) N = N^T H N v = \lambda N^T G N v = \lambda N^T G(\Delta_G) N v.$$

and thus the generalized eigenproblem (3.3), and hence eigenvalues of $K_{G(\Delta_G)}^{-1} K_{H(\Delta_H)}$, are unaltered.

In order to improve upon Theorem 3.2, now suppose that we may partition the columns of $A$ so that

$$A = (A_1 \;\; A_2),$$

and so that its leading $m$ by $m$ sub-matrix

| **A3** | $A_1$ is nonsingular; |
|---|---|

in practice, this may involve column permutations, but without loss of generality we simply assume here that any required permutations have already been carried out. Given **A3**, we shall be particularly concerned with the *reduced-space* basis matrix

$$N = \begin{pmatrix} R \\ I \end{pmatrix}, \quad \text{where} \quad R = -A_1^{-1} A_2. \tag{3.6}$$

Such basis matrices play vital roles in simplex (pivoting)-type methods for linear programming [2,23], and more generally in active-set methods for nonlinear optimization [25,35,36].

Suppose that we partition $G$ and $H$ so that

$$G = \begin{pmatrix} G_{11} & G_{21}^T \\ G_{21} & G_{22} \end{pmatrix} \text{ and } H = \begin{pmatrix} H_{11} & H_{21}^T \\ H_{21} & H_{22} \end{pmatrix}, \tag{3.7}$$

where $G_{11}$ and $H_{11}$ are (respectively) the leading $m$ by $m$ sub-matrices of $G$ and $H$. Then (3.6) and (3.7) give

$$\begin{aligned} N^T G N &= G_{22} + R^T G_{21}^T + G_{21} R + R^T G_{11} R \\ \text{and } N^T H N &= H_{22} + R^T H_{21}^T + H_{21} R + R^T H_{11} R \end{aligned}$$

In order to improve the eigenvalue distribution resulting from our attempts to precondition $K_H$ by $K_G$, we consider the consequences of picking $G$ to reproduce certain portions of $H$.

First, consider the case where

$$G_{22} = H_{22}, \text{ but } G_{11} = 0 \text{ and } G_{21} = 0. \tag{3.8}$$

**Theorem 3.4.** [17, Thm. 2.3] *Suppose that $G$ and $H$ are as in (3.7) and that (3.8) and* **A3** *hold. Suppose furthermore that $H_{22}$ is positive definite, and let*

$$\begin{aligned} \rho \stackrel{\text{def}}{=} & \min\left[\ \text{rank}(A_2), \text{rank}(H_{21})\ \right] \\ & + \min\left[\ \text{rank}(A_2), \text{rank}(H_{21}) + \min[\ \text{rank}(A_2), \text{rank}(H_{11})\ ]\ \right]. \end{aligned}$$

*Then $K_G^{-1} K_H$ has at most*

$$\text{rank}(R^T H_{21}^T + H_{21} R + R^T H_{11} R) + 1 \le \min(\rho, n-m) + 1 \le \min(2m, n-m) + 1$$

*distinct eigenvalues.*

As we have seen from Theorem 3.3, the restriction that $H_{22}$ be positive definite is not as severe as it might first seem, particularly if we can entertain the possibility of using the positive-definite matrix $H_{22} + A_2^T \Delta A_2$ instead.

The eigenvalue situation may be improved if we consider the case where

$$G_{22} = H_{22} \text{ and } G_{11} = H_{11} \text{ but } G_{21} = 0. \tag{3.9}$$

**Theorem 3.5.** [17, Thm. 2.4] *Suppose that $G$ and $H$ are as in (3.7) and that (3.9) and* **A3** *hold. Suppose furthermore that $H_{22} + R^T H_{11}^T R$ is positive definite, and that*

$$\nu \stackrel{\text{def}}{=} 2 \min\left[\ \text{rank}(A_2), \text{rank}(H_{21})\ \right].$$

*Then $K_G^{-1} K_H$ has at most*

$$\text{rank}(R^T H_{21}^T + H_{21} R) + 1 \le \nu + 1 \le \min(2m, n-m) + 1$$

*distinct eigenvalues.*

The same is true when

$$G_{22} = H_{22} \ \text{ and } \ G_{21} = H_{21} \ \text{ but } \ G_{11} = 0. \tag{3.10}$$

**Theorem 3.6.** [17, Thm. 2.5] *Suppose that $G$ and $H$ are as in (3.7) and that (3.10) and* **A3** *hold. Suppose furthermore that $H_{22} + R^T H_{21}^T + H_{21} R$ is positive definite, and that*

$$\mu \stackrel{\text{def}}{=} \min \left[ \ \operatorname{rank}(A_2), \operatorname{rank}(H_{11}) \ \right].$$

*Then $K_G^{-1} K_H$ has at most*

$$\operatorname{rank}(R^T H_{11} R) + 1 \leq \mu + 1 \leq \min(m, n-m) + 1$$

*distinct eigenvalues.*

## 3.2 General $C$

Having obtained tighter results for the case $C = 0$ than simply implied by Theorem 3.1, we now show how these results may be applied to the general case. Suppose that we denote the coefficient matrices of the systems (2.2) and (2.9) by

$$\bar{K}_H \stackrel{\text{def}}{=} \begin{pmatrix} H & 0 & A^T \\ 0 & D^{-1} & E^T \\ A & E & 0 \end{pmatrix} \ \text{ and } \ \bar{K}_G \stackrel{\text{def}}{=} \begin{pmatrix} G & 0 & A^T \\ 0 & D^{-1} & E^T \\ A & E & 0 \end{pmatrix}$$

respectively. Recalling the definitions (2.5) and (2.7) of $H_N$ ad $G_N$, the following result is a direct consequence of Theorem 3.2.

**Corollary 3.7.** *Suppose that $N$ is any ($n$ by $n + l - m$) basis matrix for the null-space of $(A \ \ E)$. Then $\bar{K}_G^{-1} \bar{K}_H$ has $2m$ unit eigenvalues, and the remaining $n + l - m$ eigenvalues are those of the generalized eigenproblem (2.10).*

We may improve on Corollary 3.7 by applying Theorems 3.4–3.6 in our more general setting. To do so, let

$$\bar{R} = -A_1^{-1}(A_2 \ \ E),$$

and note that

$$\bar{K}_H \stackrel{\text{def}}{=} \left( \begin{array}{c|cc|c} H_{11} & H_{21}^T & 0 & A_1^T \\ \hline H_{21} & H_{22}^T & 0 & A_2^T \\ 0 & 0 & D^{-1} & E^T \\ \hline A_1 & A_2 & E & 0 \end{array} \right) \ \text{ and } \ \bar{K}_G \stackrel{\text{def}}{=} \left( \begin{array}{c|cc|c} G_{11} & G_{21}^T & 0 & A_1^T \\ \hline G_{21} & G_{22}^T & 0 & A_2^T \\ 0 & 0 & D^{-1} & E^T \\ \hline A_1 & A_2 & E & 0 \end{array} \right).$$

We then have the following immediate consequences.

**Corollary 3.8.** *Suppose that $G$ and $H$ are as in (3.7) and that (3.8) and* **A3** *hold. Suppose furthermore that*

$$\begin{pmatrix} H_{22} & 0 \\ 0 & D^{-1} \end{pmatrix} \tag{3.11}$$

*is positive definite, and let*

$$\bar{\rho} = \min\left[\ \eta, \operatorname{rank}(H_{21})\ \right] + \min\left[\ \eta, \operatorname{rank}(H_{21}) + \min[\eta, \operatorname{rank}(H_{11})]\ \right],$$

*where $\eta = \operatorname{rank}(A_2 \ \ E)$. Then $\bar{K}_G^{-1}\bar{K}_H$ has at most*

$$\operatorname{rank}(\bar{R}^T H_{21}^T + H_{21}\bar{R} + \bar{R}^T H_{11}\bar{R}) + 1 \le \min(\bar{\rho}, n + l - m) + 1 \le \min(2m, n + l - m) + 1$$

*distinct eigenvalues.*

**Corollary 3.9.** *Suppose that $G$ and $H$ are as in (3.7) and that (3.9) and* **A3** *hold. Suppose furthermore that*

$$\begin{pmatrix} H_{22} & 0 \\ 0 & D^{-1} \end{pmatrix} + \bar{R}^T H_{11}^T \bar{R} \tag{3.12}$$

*is positive definite, and that*

$$\bar{\nu} = 2\min\left[\ \eta, \operatorname{rank}(H_{21})\ \right],$$

*where $\eta = \operatorname{rank}(A_2 \ \ E)$. Then $\bar{K}_G^{-1}\bar{K}_H$ has at most*

$$\operatorname{rank}(\bar{R}^T H_{21}^T + H_{21}\bar{R}) + 1 \le \bar{\nu} + 1 \le \min(2m, n + l - m) + 1$$

*distinct eigenvalues.*

**Corollary 3.10.** *Suppose that $G$ and $H$ are as in (3.7) and that (3.10) and* **A3** *hold. Suppose furthermore that*

$$\begin{pmatrix} H_{22} & 0 \\ 0 & D^{-1} \end{pmatrix} + \bar{R}^T H_{21}^T + H_{21}\bar{R} \tag{3.13}$$

*is positive definite, and that*

$$\bar{\mu} = \min\left[\ \eta, \operatorname{rank}(H_{11})\ \right],$$

*where $\eta = \operatorname{rank}(A_2 \ \ E)$. Then $\bar{K}_G^{-1}\bar{K}_H$ has at most*

$$\operatorname{rank}(\bar{R}^T H_{11}\bar{R}) + 1 \le \bar{\mu} + 1 \le \min(m, n + l - m) + 1$$

*distinct eigenvalues.*

While the requirements that (3.11)–(3.13) be positive definite may at first seem strong assumptions, as before this is not as severe as it might first seem, for we have the following immediate corollary to Theorem 3.3.

**Corollary 3.11.** *The inertial requirement (2.6) holds for a given $H$ if and only if there exists a positive semi-definite matrix $\bar{\Delta}$ such that*

$$\begin{pmatrix} H & 0 \\ 0 & D^{-1} \end{pmatrix} + \begin{pmatrix} A^T \\ E^T \end{pmatrix} \Delta (A \;\; E)$$

*is positive definite for all $\Delta$ for which $\Delta - \bar{\Delta}$ is positive semi-definite. In particular, if (2.6) holds, $H + A^T \Delta A$ and $E^T \Delta E + D^{-1}$ are positive definite for all such $\Delta$.*

Just as we did for (3.4)–(3.5), we may rewrite (2.2) as the equivalent

$$\begin{pmatrix} H + A^T \Delta A & A^T \Delta E & A^T \\ E^T \Delta A & E^T \Delta E + D^{-1} & E^T \\ A & E & 0 \end{pmatrix} \begin{pmatrix} x \\ z \\ w \end{pmatrix} = \begin{pmatrix} g \\ 0 \\ 0 \end{pmatrix},$$

where $w = y - \Delta(Ax + Ez) = y - \Delta(Ax - Cy) = y$. Eliminating the variable $z$, we find that

$$\begin{pmatrix} H + A^T \Delta A & A^T P^T \\ PA & -W \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = - \begin{pmatrix} g \\ 0 \end{pmatrix},$$

where

$$P = I - \Delta W \;\text{ and }\; W = E(E^T \Delta E + D^{-1})^{-1} E^T.$$

Hence

$$\begin{pmatrix} H + A^T \Delta A & A^T \\ A & -\bar{C} \end{pmatrix} \begin{pmatrix} x \\ \bar{y} \end{pmatrix} = - \begin{pmatrix} g \\ 0 \end{pmatrix}, \tag{3.14}$$

where

$$\bar{C} = P^{-1} W P^{-T} = (I - \Delta W)^{-1} W (I - W\Delta)^{-1} \;\text{ and }\; \bar{y} = P^T y. \tag{3.15}$$

Thus it follows from Corollary 3.11 that we may rewrite (2.2) so that its trailing and leading diagonal blocks are, respectively, negative semi- and positive definite. If we are prepared to tolerate fill-in in these blocks, requirements (3.11)–(3.13) then seem more reasonable.

Although (3.15) may appear complicated for general $C$, $\bar{C}$ is diagonal whenever $C$ is. More generally, if $E = I$, $\bar{C} = D + D\Delta D$ and we may recover $y = (I + \Delta D)\bar{y}$.

# 4 Suitable preconditioners

It has long been common practice (at least in optimization circles) [3,6,12,21,24,34,39,44] to use *explicit-factorization* preconditioners of the form (1.2) by specifying $G$ and factorizing $M_G$ using a suitable symmetric, indefinite package such as `MA27` [20] or `MA57` [19]. While such techniques have often been successful, they have usually been rather *ad hoc*, with little attempt to improve upon the eigenvalue distributions beyond those suggested by Theorem 3.1. In this section we investigate an implicit-factorization alternative.

## 4.1 Implicit-factorization preconditioners

Recently Dollar and Wathen [18] proposed a class of incomplete factorizations for saddle-point problems ($C = 0$), based upon earlier work by Schilders [42]. They consider preconditioners of the form

$$M_G = PBP^T, \tag{4.1}$$

where solutions with each of the matrices $P$, $B$ and $P^T$ are easily obtained. In particular, rather than obtaining $P$ and $B$ from a given $M_G$, *$M_G$ is derived from specially chosen $P$ and $B$.* In this section, we examine a broad class of methods of this form.

In order for the methods we propose to be effective, we shall require that

**A4** $A_1$ and its transpose are easily invertible.

Since there is considerable flexibility in choosing the "basis" $A_1$ from the rectangular matrix $A$ by suitable column interchanges, assumption **A4** is often easily, and sometimes trivially, satisfied. Note that the problem of determining the "sparsest" $A_1$ is NP hard, [9,10], while numerical considerations must be given to ensure that $A_1$ is not badly conditioned if at all possible [25]. More generally, we do not necessarily assume that $A_1$ is sparse or has a sparse factorization, merely that there are effective ways to solve systems involving $A_1$ and $A_1^T$. For example, for many problems involving constraints arising from the discretization of partial differential equations, there are highly effective *iterative* methods for such systems [4].

Suppose that

$$P = \begin{pmatrix} P_{11} & P_{12} & A_1^T \\ P_{21} & P_{22} & A_2^T \\ P_{31} & P_{32} & P_{33} \end{pmatrix} \text{ and } B = \begin{pmatrix} B_{11} & B_{21}^T & B_{31}^T \\ B_{21} & B_{22} & B_{32}^T \\ B_{31} & B_{32} & B_{33} \end{pmatrix}. \tag{4.2}$$

Our goal is to ensure that

$$(M_G)_{31} = A_1, \tag{4.3a}$$
$$(M_G)_{32} = A_2 \tag{4.3b}$$
$$\text{and } (M_G)_{33} = -C, \tag{4.3c}$$

whenever $M_G = PBP^T$. Pragmatically, though, we are only interested in the case where one of the three possibilities

$$P_{11} = 0,\ P_{12} = 0 \text{ and } P_{32} = 0, \tag{4.4a}$$
$$\text{or } P_{11} = 0,\ P_{12} = 0 \text{ and } P_{21} = 0, \tag{4.4b}$$
$$\text{or } P_{12} = 0,\ P_{32} = 0 \text{ and } P_{33} = 0 \tag{4.4c}$$

(as well as non-singular $P_{31}$ and $P_{22}$) hold, since only then will $P$ be easily block-invertible. Likewise, we restrict ourselves to the three general cases

$$B_{21} = 0,\ B_{31} = 0 \text{ and } B_{32} = 0 \text{ with easily invertible } B_{11}, B_{22} \text{ and } B_{33}, \tag{4.5a}$$
$$B_{32} = 0 \text{ and } B_{33} = 0 \text{ with easily invertible } B_{31} \text{ and } B_{22}, \tag{4.5b}$$
$$\text{or } B_{11} = 0 \text{ and } B_{21} = 0 \text{ with easily invertible } B_{31} \text{ and } B_{22}, \tag{4.5c}$$

so that $B$ is block-invertible. $B$ is also easily block invertible if

$$B_{21} = 0 \text{ and } B_{32} = 0 \text{ with easily invertible } \begin{pmatrix} B_{11} & B_{31}^T \\ B_{31} & B_{33} \end{pmatrix} \text{ and } B_{22}, \tag{4.6}$$

and we will also consider this possibility.

We consider all of these possibilities in detail in Appendix A, and summarize our findings in Tables 4.1 and 4.2. We have identified eleven possible classes of easily-invertible factors that are capable of reproducing the $A$ and $C$ blocks of $M_G$, a further two which may be useful when $C$ is diagonal, and one that is only applicable if $C = 0$.

Notice that there are *never* restrictions on $P_{22}$ and $B_{22}$.

## 4.2 Reproducing $H$

Having described families of preconditioners which are capable of reproducing the required components $A$ and $C$ of $M_G$, we now examine what form the resulting $G$ takes. In particular, we consider which sub-matrices of $G$ can be defined to completely reproduce the associated sub-matrix of $H$; we say that a component $G_{ij}$, $i, j \in \{1, 2\}$, is *complete* if it is possible to choose it so that $G_{ij} = H_{ij}$. We give the details in Appendix B, and summarize our findings for each of the 14 families from Section 4.1 in Table 4.3. In Table 4.3 the superscript $^1$ indicates that the value of $G_{21}$ is dependent on the choice of $G_{11}$. If $G_{ij}$, $i, j \in \{1, 2\}$, is a zero matrix, then a superscript $^2$ is used. The superscript $^3$ means that $G_{21}$ is dependent on the choice of $G_{11}$ when $C = 0$, but complete otherwise, whilst the superscript $^4$ indicates that $G_{11}$ is only guaranteed to be complete when $C = 0$.

Some of the sub-matrices in the factors $P$ and $B$ can be arbitrarily chosen without changing the completeness of the family. We shall call these "free blocks." For example, consider Family 2 from Table 4.1. The matrix $G$ produced by this family always satisfies $G_{11} = 0$, $G_{21} = 0$, and $G_{22} = P_{22} B_{22} P_{22}^T$. Hence, $P_{22}$ can be defined as any non-singular matrix of suitable dimension, and $B_{22}^T$ can be subsequently chosen so that $G_{22} = H_{22}$. The simplest choice for $P_{22}$ is the identity matrix. We observe, that the choice of the remaining sub-matrices in $P$ and $B$ will not affect the completeness of the factorization, and are only required to satisfy the conditions given in Table 4.1. The simplest choices for these sub-matrices will be $P_{31} = I$, and $B_{11} = 0$, giving $P_{33} = -\frac{1}{2}C$, and $B_{31} = I$. Using these simple choices we obtain:

$$P = \begin{pmatrix} 0 & 0 & A_1^T \\ 0 & I & A_2^T \\ I & 0 & -\frac{1}{2}C \end{pmatrix} \text{ and } B = \begin{pmatrix} 0 & 0 & I \\ 0 & B_{22} & 0 \\ I & 0 & 0 \end{pmatrix}.$$

The simplest choice of the free blocks may result in some of the families having the same factors as other families. This is indicated in the Comments column of the table. Table 4.3 also gives the conditions that $C$ must satisfy to use the family, and whether the family is feasible to use, i.e., are the conditions on the blocks given in Tables 4.1 and 4.2 easily satisfied?

| Family/ reference | $P$ | $B$ | conditions |
|---|---|---|---|
| 1. (A.3)–(A.4) | $\begin{pmatrix} 0 & 0 & A_1^T \\ 0 & P_{22} & A_2^T \\ P_{31} & 0 & P_{33} \end{pmatrix}$ | $\begin{pmatrix} B_{11} & 0 & 0 \\ 0 & B_{22} & 0 \\ 0 & 0 & B_{33} \end{pmatrix}$ | $B_{11} = -P_{31}^{-1}(C + P_{33})P_{31}^{-T}$ <br> $B_{33} = P_{33}^{-1}$ |
| 2. (A.10)–(A.11) | $\begin{pmatrix} 0 & 0 & A_1^T \\ 0 & P_{22} & A_2^T \\ P_{31} & 0 & P_{33} \end{pmatrix}$ | $\begin{pmatrix} B_{11} & 0 & B_{31}^T \\ 0 & B_{22} & 0 \\ B_{31} & 0 & 0 \end{pmatrix}$ | $P_{31} = B_{31}^{-T}$ <br> $P_{33} + P_{33}^T + P_{31}B_{11}P_{31}^T = -C$ |
| 3. (A.12) | $\begin{pmatrix} 0 & 0 & A_1^T \\ P_{21} & P_{22} & A_2^T \\ P_{31} & 0 & -C \end{pmatrix}$ | $\begin{pmatrix} B_{11} & 0 & B_{31}^T \\ 0 & B_{22} & 0 \\ B_{31} & 0 & 0 \end{pmatrix}$ | $B_{31} = P_{31}^{-T}$ <br> $B_{11} = P_{31}^{-1}CP_{31}^{-T}$ |
| 4. (A.16)–(A.17) | $\begin{pmatrix} 0 & 0 & A_1^T \\ P_{21} & P_{22} & A_2^T \\ P_{31} & 0 & P_{33} \end{pmatrix}$ | $\begin{pmatrix} 0 & 0 & B_{31}^T \\ 0 & B_{22} & B_{32}^T \\ B_{31} & B_{32} & 0 \end{pmatrix}$ | $P_{21} = -P_{22}B_{32}^TB_{31}^{-T}$ <br> $P_{31} = B_{31}^{-T}$ <br> $P_{33} + P_{33}^T = -C$ |
| 5. (A.18)–(A.19) | $\begin{pmatrix} 0 & 0 & A_1^T \\ P_{21} & P_{22} & A_2^T \\ P_{31} & 0 & P_{33} \end{pmatrix}$ | $\begin{pmatrix} 0 & 0 & B_{31}^T \\ 0 & B_{22} & B_{32}^T \\ B_{31} & B_{32} & B_{33} \end{pmatrix}$ | $-C = P_{33} + P_{33}^T - P_{33}B_{33}P_{33}^T$ <br> $B_{31} = (I - B_{33}P_{33}^T)P_{31}^{-T}$ <br> $B_{32} = -B_{31}P_{21}^TP_{22}^{-T}$ |
| 6. (A.20)–(A.21) | $\begin{pmatrix} 0 & 0 & A_1^T \\ 0 & P_{22} & A_2^T \\ P_{31} & P_{32} & P_{33} \end{pmatrix}$ | $\begin{pmatrix} B_{11} & B_{21}^T & B_{31}^T \\ B_{21} & B_{22} & 0 \\ B_{31} & 0 & 0 \end{pmatrix}$ | $P_{31} = B_{31}^{-T}$ <br> $P_{32} = -P_{31}B_{21}^TB_{22}^{-1}$ <br> $P_{33} + P_{33}^T$ <br> $= -C - P_{31}(B_{11} - B_{21}^TB_{22}^{-1}B_{21})P_{31}^T$ |
| 7. (A.28)–(A.29) | $\begin{pmatrix} 0 & 0 & A_1^T \\ 0 & P_{22} & A_2^T \\ P_{31} & P_{32} & P_{33} \end{pmatrix}$ | $\begin{pmatrix} 0 & 0 & B_{31}^T \\ 0 & B_{22} & B_{32}^T \\ B_{31} & B_{32} & B_{33} \end{pmatrix}$ | $P_{33} + P_{33}^T + P_{33}(B_{33} - B_{32}B_{22}^{-1}B_{32}^T)P_{33}^T$ <br> $= -C$ <br> $P_{32} = -P_{33}B_{32}B_{22}^{-1}$ <br> $P_{31} = (I - P_{32}B_{32}^T - P_{33}B_{33}^T)B_{31}^{-T}$ |

Table 4.1: Possible implicit factors for the preconditioner (1.2). We give the $P$ and $B$ factors and any necessary restrictions on their entries. We also associate a family number with each class of implicit factors, and indicate where each is derived in Appendix A.

| Family/ reference | $P$ | $B$ | conditions |
|---|---|---|---|
| 8. (A.30) | $\begin{pmatrix} A_1^T & 0 & A_1^T \\ A_2^T & P_{22} & A_2^T \\ -C & 0 & 0 \end{pmatrix}$ | $\begin{pmatrix} -C^{-1} & 0 & 0 \\ 0 & B_{22} & 0 \\ 0 & 0 & B_{33} \end{pmatrix}$ | $C$ invertible |
| 9. (A.31)– (A.32) | $\begin{pmatrix} P_{11} & 0 & A_1^T \\ P_{21} & P_{22} & A_2^T \\ P_{31} & 0 & 0 \end{pmatrix}$ | $\begin{pmatrix} B_{11} & B_{21}^T & B_{31}^T \\ B_{21} & B_{22} & 0 \\ B_{31} & 0 & 0 \end{pmatrix}$ | $B_{11} = -P_{31}^{-1} C P_{31}^{-T}$<br>$B_{31} = P_{31}^{-T} - M B_{11}$<br>$B_{21} = P_{22}^{-1}(P_{21} - A_2^T M) B_{11}$<br>$P_{11} = A_1^T M$ for some invertible $M$ |
| 10. (A.33) | $\begin{pmatrix} P_{11} & 0 & A_1^T \\ P_{21} & P_{22} & A_2^T \\ P_{31} & 0 & 0 \end{pmatrix}$ | $\begin{pmatrix} 0 & 0 & B_{31}^T \\ 0 & B_{22} & B_{32}^T \\ B_{31} & B_{32} & B_{33} \end{pmatrix}$ | $C = 0$<br>$P_{31} = B_{31}^{-T}$ |
| 11. (A.37)– (A.38) | $\begin{pmatrix} 0 & 0 & A_1^T \\ P_{21} & P_{22} & A_2^T \\ P_{31} & 0 & -C \end{pmatrix}$ | $\begin{pmatrix} B_{11} & 0 & B_{31}^T \\ 0 & B_{22} & 0 \\ B_{31} & 0 & B_{33} \end{pmatrix}$ | $C$ invertible<br>$P_{31}^T = B_{11}^{-1} B_{31}^T C$<br>$B_{33} = (B_{31} P_{31}^T - I) C^{-1}$ |
| 12. (A.37), (A.42)– (A.43) | $\begin{pmatrix} 0 & 0 & A_1^T \\ P_{21} & P_{22} & A_2^T \\ P_{31} & 0 & -C \end{pmatrix}$ | $\begin{pmatrix} B_{11} & 0 & B_{31}^T \\ 0 & B_{22} & 0 \\ B_{31} & 0 & B_{33} \end{pmatrix}$ | $B_{11} = P_{31}^{-1} C P_{31}^{-T}$<br>$B_{31} = P_{31}^{-T}$, where<br>$B_{33} C = 0$ |
| 13. (A.44)– (A.45) | $\begin{pmatrix} 0 & 0 & A_1^T \\ 0 & P_{22} & A_2^T \\ P_{31} & 0 & P_{33} \end{pmatrix}$ | $\begin{pmatrix} B_{11} & 0 & B_{31}^T \\ 0 & B_{22} & 0 \\ B_{31} & 0 & B_{33} \end{pmatrix}$ | $P_{31} = (I - P_{33} B_{33}) B_{31}^{-T}$<br>$B_{11} = P_{31}^{-1} \left( P_{33} B_{33} P_{33}^T - C - P_{33} - P_{33}^T \right) P_{31}^{-T}$ |
| 14. (A.47)– (A.48) | $\begin{pmatrix} P_{11} & 0 & A_1^T \\ P_{21} & P_{22} & A_2^T \\ P_{31} & 0 & 0 \end{pmatrix}$ | $\begin{pmatrix} B_{11} & 0 & B_{31}^T \\ 0 & B_{22} & 0 \\ B_{31} & 0 & B_{33} \end{pmatrix}$ | $B_{11} = -P_{31}^{-1} C P_{31}^{-T}$<br>$B_{31} = P_{31}^{-T} - M B_{11}$<br>$P_{11} = A_1^T M$<br>$P_{21} = A_2^T M$ for some invertible $M$ |

Table 4.2: Possible implicit factors for the preconditioner (1.2) (continued). We give the $P$ and $B$ factors and any necessary restrictions on their entries. We also associate a family number with each class of implicit factors, and indicate where each is derived in Appendix A.

Table 4.4 gives some guidance towards which families from Tables 4.1 and 4.2 should be used in the various cases of $G$ given in Section 3. We also suggest simple choices for the free blocks. In our view, although Table 4.3 indicates that it is theoretically possible to reproduce all of $H$ using (e.g.) Family 9, in practice this is unviable because of the density of the matrices that need to be factorized.

| Family | Completeness | | | Conditions on $C$ | Feasible to use | Comments |
|---|---|---|---|---|---|---|
| | $G_{11}$ | $G_{21}$ | $G_{22}$ | | | |
| 1. | ✓ | $\times^1$ | ✓ | any $C$ | ✓ | |
| 2. | $\times^2$ | $\times^2$ | ✓ | any $C$ | ✓ | |
| 3. | $\times^2$ | ✓ | ✓ | any $C$ | ✓ | |
| 4. | $\times^2$ | $\times^2$ | ✓ | any $C$ | ✓ | Simplest choice of "free-blocks" is the same as that for Family 2. |
| 5. | ✓ | $\times^1$ | ✓ | any $C$ | $C = 0$ | |
| 6. | $\times^2$ | $\times^2$ | ✓ | any $C$ | ✓ | Simplest choice of "free-blocks" is the same as that for Family 2. |
| 7. | ✓ | $✓^3$ | ✓ | any $C$ | $C = 0$ | If $C = 0$ and use simplest choice of "free-blocks", then same as that for Family 5 with $C = 0$. |
| 8. | ✓ | $\times^1$ | ✓ | non-singular | ✓ | |
| 9. | ✓ | ✓ | ✓ | any $C$ | $C = 0$ | |
| 10. | ✓ | ✓ | ✓ | $C = 0$ | ✓ | Generalization of factorization suggested by Schilders, [18, 42]. |
| 11. | ✓ | ✓ | ✓ | non-singular | ✓ | |
| 12. | $✓^4$ | ✓ | ✓ | any $C$ | diagonal $C$ | $C = 0$ gives example of Family 10. $C$ non-singular gives Family 3. |
| 13. | ✓ | $\times^1$ | ✓ | any $C$ | ✓ | |
| 14. | ✓ | $\times^1$ | ✓ | any $C$ | ✓ | $C = 0$ gives example of Family 10. |

Table 4.3: Blocks of $G$ for the families of preconditioners given in Tables 4.1 and 4.2.

| Sub-blocks of $G$ | Conditions on $C$ | Family | Free block choices |
|---|---|---|---|
| $G_{22} = H_{22}$, $G_{11} = 0$, $G_{21} = 0$ | any $C$ | 2 | $P_{22} = I$, $P_{31} = I$, $B_{11} = 0$ |
| $G_{22} = H_{22}$, $G_{11} = H_{11}$, $G_{21} = 0$ | $C = 0$ | 10 | $B_{21} = 0$, $P_{22} = I$, $P_{31} = I$ |
| $G_{22} = H_{22}$, $G_{11} = H_{11}$, $G_{21} = 0$ | $C$ non-singular | 11 | $P_{22} = I$, $P_{31} = I$ |
| $G_{22} = H_{22}$, $G_{21} = H_{21}$, $G_{11} = 0$ | any $C$ | 3 | $P_{22} = I$, $P_{31} = I$ |

Table 4.4: Guidance towards which family to use to generate the various choices of $G$ given in Section 3.

# 5 Numerical Examples

In this section we examine how effective implicit factorization preconditioners might be when compared with explicit factorization ones. We consider problems generated using the complete set of quadratic programming examples from the CUTEr [31] test set used in our previous experiments for the $C = 0$ case [17]. All inequality constraints are converted to equations by adding slack variables, and a suitable "barrier" penalty term is added to the diagonal of the Hessian for each bounded or slack variable to simulate systems that might arise during an iteration of an interior-point method for such problems; in each of the test problems the value 1.1 is used. The resulting equality-constrained quadratic programs are then of the form

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \;\; g^T x + \tfrac{1}{2} x^T H x \;\; \text{subject to} \;\; Ax = 0. \tag{5.7}$$

Given this data $H$ and $A$, two illustrative choices of diagonal $C$ are considered, namely

$$c_{ii} = 1 \;\; \text{for} \;\; 1 \leq i \leq m, \tag{5.8}$$

and

$$c_{ii} = \begin{cases} 0 & \text{for} \;\; 1 \leq i \leq \lceil \frac{m}{2} \rceil \\ 1 & \text{for} \;\; \lceil \frac{m}{2} \rceil + 1 \leq i \leq m; \end{cases} \tag{5.9}$$

in practice such $C$ may be thought of as regularization terms for some or all on the constraints in (5.7). Our aim is thus to solve the system (1.1) using a suitably preconditioned PPCG iteration.

Rather than present large tables of data (which we defer to Appendix C), here we use performance profiles [14] to illustrate our results. To explain the idea, let $\mathcal{P}$ represent the set of preconditioners that we wish to compare. Suppose that the run of PPCG using a given preconditioner $i \in \mathcal{P}$ reports the total CPU time $t_{ij} \geq 0$ when executed on example $j$ from the test set $\mathcal{T}$. For all problems $j \in \mathcal{T}$, we want to compare the performance of algorithm $i$ with the performance of the fastest algorithm in the set $\mathcal{P}$. For $j \in \mathcal{T}$, let

$t_j^{\text{MIN}} = \min\{t_{ij}; i \in \mathcal{P}\}$. Then for $\alpha \geq 1$ and each $i \in \mathcal{P}$ we define

$$k(t_{ij}, t_j^{\text{MIN}}, \alpha) = \begin{cases} 1 & \text{if } t_{ij} \leq \alpha t_j^{\text{MIN}} \\ 0 & \text{otherwise.} \end{cases}$$

The *performance profile* [14] of algorithm $i$ is then given by the function

$$p_i(\alpha) = \frac{\sum_{j \in \mathcal{T}} k(t_{ij}, t_j^{\text{MIN}}, \alpha)}{|\mathcal{T}|}, \quad \alpha \geq 1.$$

Thus $p_i(1)$ gives the fraction of the examples for which algorithm $i$ is the most effective (according to the statistic $t_{ij}$), $p_i(2)$ gives the fraction for which algorithm $i$ is within a factor of 2 of the best, and $\lim_{\alpha \to \infty} p_i(\alpha)$ gives the fraction for which the algorithm succeeded.

We consider two explicit factorization preconditioners, one using exact factors ($G = H$), and the other using a simple projection ($G = I$). A Matlab interface to the HSL package MA57 [19] (version 2.2.1) is used to factorize $K_G$ and subsequently solve (3.4). Three implicit factorizations of the form (4.1) with factors (4.2) are also considered. The first is from Family 1 (Table 4.1), and aims for simplicity by choosing $P_{31} = I$, $P_{33} = I = B_{33}$ and $B_{22} = I = P_{22}$, and this leads $B_{11} = -(C+I)$; such a choice does not necessarily reproduce any of $H$, but is inexpensive to use. The remaining implicit factorizations are from Family 2 (Table 4.1). The former (marked (a) in the Figures) selects $G_{22} = H_{22}$ while the latter (marked (b) in the Figures) chooses $G_{22} = I$; for simplicity we chose $P_{31} = I = B_{31}$, $B_{11} = 0$, $P_{22} = I$ and $P_{33} = -\frac{1}{2}C$ (see §4.2), and thus we merely require that $B_{22} = H_{22}$ for case (a) and $B_{22} = I$ for case (b)—we use MA57 to factorize $H_{22}$ in the former case.

Given $A$, a suitable basis matrix $A_1$ is found by finding a sparse LU factorization of $A^T$ using the built-in Matlab function lu. An attempt to correctly identify rank is controlled by tight threshold column pivoting, in which any pivot may not be smaller than a factor $\tau = 2$ of the largest entry in its (uneliminated) column [25, 26]. The rank is estimated as the number of pivots, $\rho(A)$, completed before the remaining uneliminated sub-matrix is judged to be numerically zero, and the indices of the $\rho(A)$ pivotal rows and columns of $A$ define $A_1$—if $\rho(A) < m$, the remaining rows of $A$ are judged to be dependent, and are discarded. Although such a strategy may not be as robust as, say, a singular-value decomposition or a QR factorization with pivoting, both our and others' experience [25] indicate it to be remarkably reliable and successful in practice. Having found $A_1$, the factors are discarded, and a fresh LU decomposition of $A_1$, with a looser threshold column pivoting factor $\tau = 100$, is computed using lu in order to try to encourage sparse factors.

All of our experiments were performed using a dual processor Intel Xeon 3.2GHz Workstation with hyper-threading and 2 Gbytes of RAM. Our codes were written and executed in Matlab 7.0 Service Pack 1.

In Figures 5.1–5.2 (see Tables C.1 and C.2 for the raw data), we compare our five preconditioning strategies for (approximately) solving the problem (1.1) when $C$ is given by (5.8) using the PPCG scheme (variant 2) described in Section 2. We consider both low and high(er) accuracy solutions. For the former, we terminate as soon as the residual $\sigma$ has been reduced more than $10^{-2}$ from its original value, while the latter requires a $10^{-8}$ reduction; these are intended to simulate the levels of accuracy that might be required within a nonlinear equation or optimization solver in early (global) and later (asymptotic) phases of the solution process.

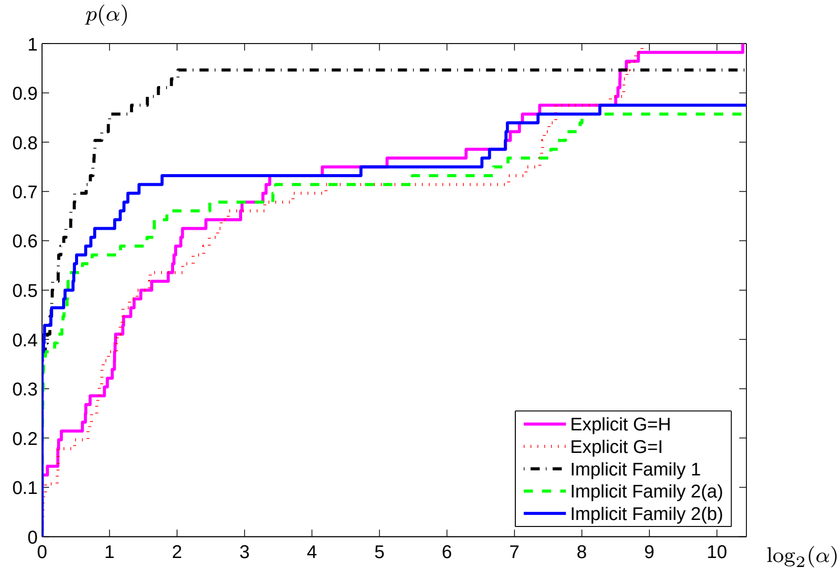


Figure 5.1: Performance profile, $p(\alpha)$: CPU time (seconds) to reduce relative residual by $10^{-2}$, when $C$ is given by (5.8).

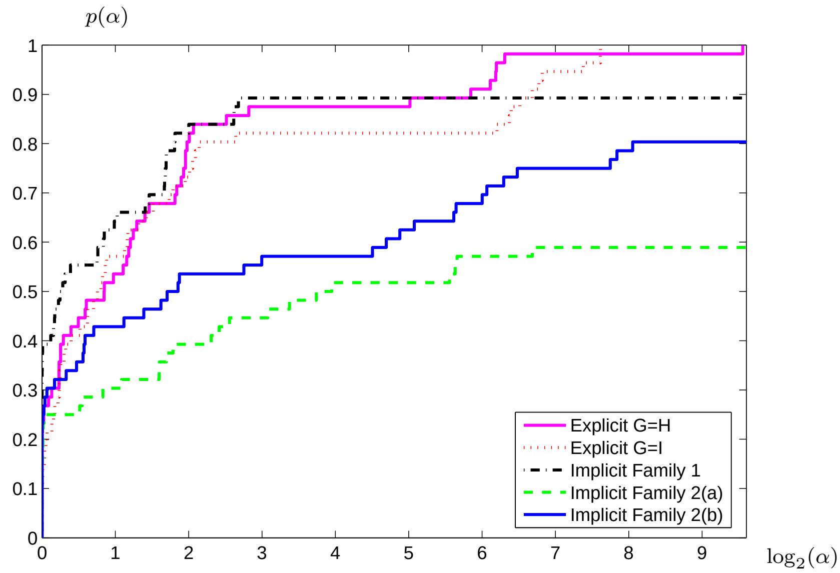


Figure 5.2: Performance profile, $p(\alpha)$: CPU time (seconds) to reduce relative residual by $10^{-8}$, when $C$ is given by (5.8).

We see that if low accuracy solutions suffice, the implicit factorizations appear to be significantly more effective at reducing the residual than their explicit counterparts. In particular, the implicit factorization from Family 1 seems to be the most effective. Of interest is that for Family 2, the cost of applying the more accurate implicit factorization that reproduces $H_{22}$ generally does not pay off relative to the cost of the cheaper implicit factorizations. For higher accuracy solutions, the leading implicit factorization still slightly outperforms the explicit factors, although now the remaining implicit factorizations are less effective.

Figures 5.3–5.4 (c.f., Tables C.3–C.4) repeat the experiments when $C$ is given by (5.9).

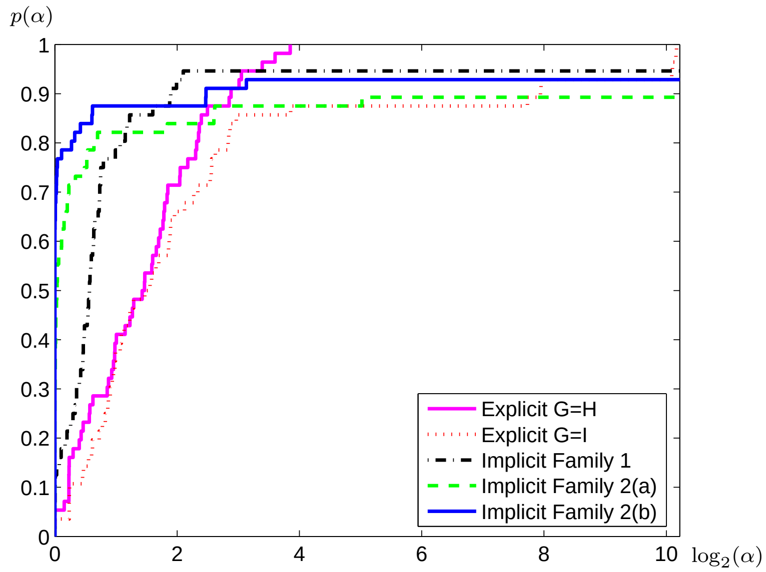


Figure 5.3: Performance profile, $p(\alpha)$: CPU time (seconds) to reduce relative residual by $10^{-2}$, when $C$ is given by (5.9).

Once again the implicit factorizations seem very effective, with a shift now to favour those from Family 2, most especially the less sophisticated of these.

# 6 Comments and conclusions

In this paper we have considered conjugate-gradient like methods for block symmetric indefinite linear systems that arise from (perturbations of) saddle point problems. Such methods require preconditioners that preserve certain sub-blocks from the original systems but allow considerable flexibility for the remaining "non-crucial" blocks. To this end, we have constructed fourteen families of implicit factorizations that are capable of reproducing the required sub-blocks and (some) of the remainder. These generalize known implicit factorizations [17, 18] for the $C = 0$ case. Improved eigenvalue clustering is possible if

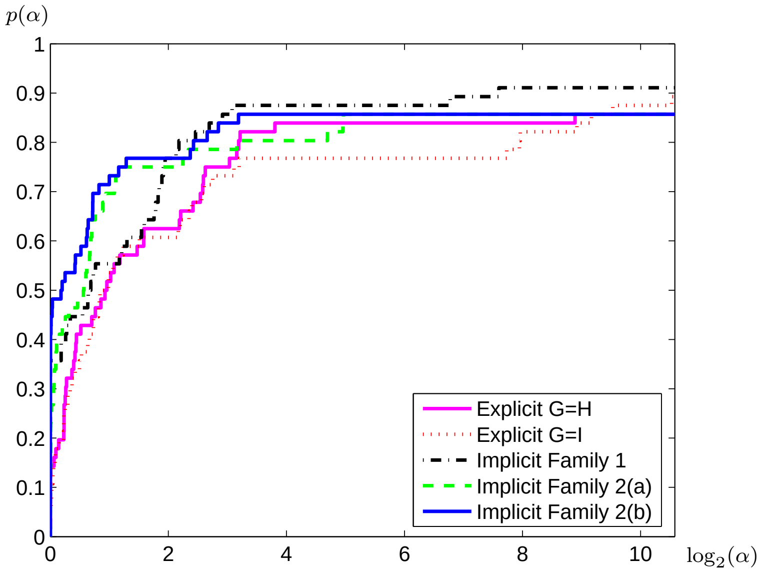


Figure 5.4: Performance profile, $p(\alpha)$: CPU time (seconds) to reduce relative residual by $10^{-8}$, when $C$ is given by (5.9).

additionally some of the "non-crucial" blocks are reproduced. We have shown numerically that these implicit-factorization preconditioners can be effective.

A number of important issues remain. Firstly, we have made no effort to find the best preconditioner(s) from amongst our families, and indeed in most cases have not even tried them in practice. As always with preconditioning, there is a delicate balance between improving clustering of eigenvalues and the cost of doing so, especially since in many applications low accuracy estimates of solution suffice. We expect promising candidates to emerge in due course, but feel it is beyond the scope of this paper to indicate more than (as we have already demonstrated) that this is a promising approach.

Secondly and as we pointed out in [17], the choice of the matrix $A_1$ is crucial, and considerations of both its stability and sparsity, and of its effect on the which of the "non-crucial" blocks may be reproduced, are vital. Thirdly (and possibly related), when experimenting with Family 3 (Table 4.1), we found that some very badly conditioned preconditioners were generated. Specifically, our aim had been to reproduce $G_{21} = H_{21}$, and for simplicity we had chosen $P_{31} = I = B_{31}$ and $B_{22} = I = P_{22}$, and this leads to $P_{21} = H_{21}A_1^{-1}$. Note that we did not try to impose additionally that $G_{22} = H_{22}$ as this would have lead to non-trivial $B_{22}$. Also notice that we did not need to form $P_{21}$, merely to operate with it (and its transpose) on given vectors. On examining the spectrum of (3.3) for some small badly conditioned examples, the preconditioner appeared to have worsened rather than improved the range of the eigenvalues. Whether this is a consequence or requiring two solves with $A_1$ (and its transpose) when applying the preconditioner rather than the single solve required when not trying to reproduce $H_{21}$, and whether the same would be true for other families trying to do the same is simply conjecture at this stage. However it is certainly a cautionary warning.

## Acknowledgment

Thanks are due to Mario Arioli both for fruitful discussions on various aspects of this work and for providing us with a Matlab interface to `MA57`.

## References

[1] M. Avriel. *Nonlinear Programming: Analysis and Methods*. Prentice-Hall, Englewood Cliffs, New Jersey, 1976.

[2] R. H. Bartels and G. H. Golub. The simplex method of linear programming using lu decompositions. *Communications of the ACM*, 12:266–268, 1969.

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

[4] G. Biros and O. Ghattas. A Lagrange-Newton-Krylov-Schur method for pde-constrained optimization. *SIAG/Optimization Views-and-News*, 11(2):12–18, 2000.

[5] J. H. Bramble and J. E. Pasciak. A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems. *Mathematics of Computation*, 50:1–17, 1988.

[6] R. H. Byrd, M. E. Hribar, and J. Nocedal. An interior point algorithm for large scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 2000.

[7] Y. Chabrillac and J.-P. Crouzeix. Definiteness and semidefiniteness of quadratic forms revisited. *Linear Algebra and its Applications*, 63:283–292, 1984.

[8] T. F. Coleman. Linearly constrained optimization and projected preconditioned conjugate gradients. In J. Lewis, editor, *Proceedings of the Fifth SIAM Conference on Applied Linear Algebra*, pages 118–122, Philadelphia, USA, 1994. SIAM.

[9] T. F. Coleman and A. Pothen. The null space problem I: complexity. *SIAM Journal on Algebraic and Discrete Methods*, 7(4):527–537, 1986.

[10] T. F. Coleman and A. Pothen. The null space problem II: algorithms. *SIAM Journal on Algebraic and Discrete Methods*, 8(4):544–563, 1987.

[11] T. F. Coleman and A. Verma. A preconditioned conjugate gradient approach to linear equality constrained minimization. Technical report, Department of Computer Sciences, Cornell University, Ithaca, New York, USA, July 1998.

[12] A. R. Conn, N. I. M. Gould, D. Orban, and Ph. L. Toint. A primal-dual trust-region algorithm for non-convex nonlinear programming. *Mathematical Programming*, 87(2):215–249, 2000.

[13] G. Debreu. Definite and semidefinite quadratic forms. *Econometrica*, 20(2):295–300, 1952.

[14] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[15] H. S. Dollar. Continuity of eigenvalues when using constraint preconditioners. Working note, Oxford University Computing Laboratory, Oxford, England, 2004.

[16] H. S. Dollar. Extending constraint preconditioners for saddle-point problems. Technical Report NA-05/02, Oxford University Computing Laboratory, Oxford, England, 2005.

[17] H. S. Dollar, N. I. M. Gould, and A. J. Wathen. On implicit-factorization constraint preconditioners. Technical Report RAL-TR-2004-036, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 2004.

[18] H. S. Dollar and A. J. Wathen. Incomplete factorization constraint preconditioners for saddle point problems. Technical Report 04/01, Oxford University Computing Laboratory, Oxford, England, 2004.

[19] I. S. Duff. MA57 - a code for the solution of sparse symmetric definite and indefinite systems. *ACM Transactions on Mathematical Software*, 30(2):118–144, 2004.

[20] I. S. Duff and J. K. Reid. The multifrontal solution of indefinite sparse symmetric linear equations. *ACM Transactions on Mathematical Software*, 9(3):302–325, 1983.

[21] C. Durazzi and V. Ruggiero. Indefinitely constrained conjugate gradient method for large sparse equality and inequality constrained quadratic problems. *Numerical Linear Algebra with Applications*, 10(8):673–688, 2002.

[22] H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite-Elements and Fast Iterative Solvers: with applications in Incompressible Fluid Dynamics*. Oxford University Press, Oxford, 2005, to appear.

[23] J. J. H. Forrest and J. A. Tomlin. Updating triangular factors of the basis to maintain sparsity in the product form simplex method. *Mathematical Programming*, 2(3):263–278, 1972.

[24] P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. Saunders. Preconditioners for indefinite systems arising in optimization. *SIAM Journal on Matrix Analysis and Applications*, 13(1):292–311, 1992.

[25] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, 12(4):979–1006, 2002.

[26] P. E. Gill and M. A. Saunders. Private communication, 1999.

[27] G. H. Golub and C. Greif. On solving block-structured indefinite linear systems. *SIAM Journal on Scientific Computing*, 24(6):2076–2092, 2003.

[28] N. I. M. Gould. On practical conditions for the existence and uniqueness of solutions to the general equality quadratic-programming problem. *Mathematical Programming*, 32(1):90–99, 1985.

[29] N. I. M. Gould. Iterative methods for ill-conditioned linear systems from optimization. In G. Di Pillo and F. Giannessi, editors, *Nonlinear Optimization and Related Topics*, pages 123–142, Dordrecht, The Netherlands, 1999. Kluwer Academic Publishers.

[30] 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(4):1375–1394, 2001.

[31] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEr (and SifDec), a Constrained and Unconstrained Testing Environment, revisited. *ACM Transactions on Mathematical Software*, 29(4):373–394, 2003.

[32] C. Greif, G. H. Golub, and J. M. Varah. Augmented Lagrangian techniques for solving saddle point linear systems. Technical report, Computer Science Department, University of British Columbia, Vancouver, Canada, 2004.

[33] C. Keller, N. I. M. Gould, and A. J. Wathen. Constraint preconditioning for indefinite linear systems. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1300–1317, 2000.

[34] L. Lukšan and J. Vlček. Indefinitely preconditioned inexact Newton method for large sparse equality constrained nonlinear programming problems. *Numerical Linear Algebra with Applications*, 5(3):219–247, 1998.

[35] B. A. Murtagh and M. A. Saunders. Large-scale linearly constrained optimization. *Mathematical Programming*, 14(1):41–72, 1978.

[36] B. A. Murtagh and M. A. Saunders. A projected Lagrangian algorithm and its implementation for sparse non-linear constraints. *Mathematical Programming Studies*, 16:84–117, 1982.

[37] J. Nocedal and S. J. Wright. *Large sparse numerical optimization*. Series in Operations Research. Springer Verlag, Heidelberg, Berlin, New York, 1999.

[38] L. A. Pavarino. Preconditioned mixed spectral finite-element methods for elasticity and Stokes problems. *SIAM Journal on Scientific Computing*, 19(6):1941–1957, 1998.

[39] I. Perugia and V. Simoncini. Block-diagonal and indefinite symmetric preconditioners for mixed finite element formulations. *Numerical Linear Algebra with Applications*, 7(7-8):585–616, 2000.

[40] B. T. Polyak. The conjugate gradient method in extremal problems. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 9:94–112, 1969.

[41] M. Rozložník and V. Simoncini. Krylov subspace methods for saddle point problems with indefinite preconditioning. *SIAM Journal on Matrix Analysis and Applications*, 24(2):368–391, 2002.

[42] W. Schilders. A preconditioning technique for indefinite systems arising in electronic circuit simulation. Talk at the one-day meeting on preconditioning methods for indefinite linear systems, TU Eindhoven, December 9, 2002.

[43] W. Schilders and E. Ter Maten. *Numerical Methods in Electromagnetics (Handbook of Numerical Analysis S.)*. Elsevier, Oxford, 2005.

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

[45] T. Zhang, K. H. Law, and G. H. Golub. On the homotopy method for perturbed symmetric generalized eigenvalue problems. *SIAM Journal on Scientific Computing*, 19(5):1625–1645, 1998.

# Appendix A

We examine each of the sub-cases mentioned in Section 4 in detail. Note that for general $P$ and $B$ partitioned as in (4.2), we have

$$\begin{aligned}(M_G)_{31} = & \; (P_{31}B_{11} + P_{32}B_{21})P_{11}^T + (P_{31}B_{21}^T + P_{32}B_{22})P_{12}^T \\ & + P_{33}(B_{31}P_{11}^T + B_{32}P_{12}^T) + (P_{31}B_{31}^T + P_{32}B_{32}^T)A_1 + P_{33}B_{33}A_1, \\ (M_G)_{32} = & \; (P_{31}B_{11} + P_{32}B_{21})P_{21}^T + (P_{31}B_{21}^T + P_{32}B_{22})P_{22}^T \\ & + P_{33}(B_{31}P_{21}^T + B_{32}P_{22}^T) + (P_{31}B_{31}^T + P_{32}B_{32}^T)A_2 + P_{33}B_{33}A_2 \\ \text{and } (M_G)_{33} = & \; (P_{31}B_{11} + P_{32}B_{21})P_{31}^T + (P_{31}B_{21}^T + P_{32}B_{22})P_{32}^T \\ & + P_{33}(B_{31}P_{31}^T + B_{32}P_{32}^T) + (P_{31}B_{31}^T + P_{32}B_{32}^T)P_{33}^T + P_{33}B_{33}P_{33}^T.\end{aligned}$$

## Case 1: (4.4a) and (4.5a) hold

If (4.4a) and (4.5a) hold, $P_{31}$, $P_{22}$, $B_{11}$, $B_{22}$ and $B_{33}$ are non singular, and

$$\begin{aligned}(M_G)_{31} = & \; P_{33}B_{33}A_1, \\ (M_G)_{32} = & \; P_{33}B_{33}A_2 + P_{31}B_{11}P_{21}^T \\ \text{and } (M_G)_{33} = & \; P_{33}B_{33}P_{33}^T + P_{31}B_{11}P_{31}^T.\end{aligned}$$

In this case, requirement (4.3a) implies that

$$P_{33}B_{33} = I \tag{A.1}$$

and thus that $P_{33}$ is symmetric. The requirement (4.3b) then forces $P_{31}B_{11}P_{21}^T = 0$, and thus that

$$P_{21} = 0$$

since $P_{31}$ and $B_{11}$ are non singular. The final requirement (4.3c) is then that

$$P_{33} + P_{31}B_{11}P_{31}^T = -C. \tag{A.2}$$

Thus, in this case,

$$P = \begin{pmatrix} 0 & 0 & A_1^T \\ 0 & P_{22} & A_2^T \\ P_{31} & 0 & P_{33} \end{pmatrix} \text{ and } B = \begin{pmatrix} B_{11} & 0 & 0 \\ 0 & B_{22} & 0 \\ 0 & 0 & B_{33} \end{pmatrix}, \tag{A.3}$$

where

$$B_{11} = -P_{31}^{-1}(C + P_{33})P_{31}^{-T} \text{ and } B_{33} = P_{33}^{-1}. \tag{A.4}$$

## Case 2: (4.4a) and (4.5b) hold

If (4.4a) and (4.5b) hold, $P_{31}$, $P_{22}$, $B_{31}$ and $B_{22}$ are non singular, and

$$\begin{aligned}(M_G)_{31} = & \; P_{31}B_{31}^TA_1, \\ (M_G)_{32} = & \; P_{31}B_{31}^TA_2 + P_{31}B_{11}P_{21}^T + P_{31}B_{21}^TP_{22}^T + P_{33}B_{31}P_{21}^T \\ \text{and } (M_G)_{33} = & \; P_{31}B_{31}^TP_{33}^T + P_{31}B_{11}P_{31}^T + P_{33}B_{31}P_{31}^T.\end{aligned}$$

In this case, requirement (4.3a) implies that

$$P_{31}B_{31}^T = I, \tag{A.5}$$

holds. It then follows from (A.5) that

$$P_{33} + P_{33}^T + P_{31}B_{11}P_{31}^T = -C \tag{A.6}$$

since we require (4.3c). The remaining requirement (4.3b) implies that $P_{31}B_{11}P_{21}^T + P_{31}B_{21}^T P_{22}^T + P_{33}B_{31}P_{21}^T = 0$, which is most easily guaranteed if either

$$B_{21} = 0 \text{ and } P_{21} = 0 \tag{A.7}$$

or

$$B_{21} = 0 \text{ and } P_{31}B_{11} = -P_{33}B_{31} \tag{A.8}$$

or

$$B_{21} = 0, \;\; B_{11} = 0 \text{ and } P_{33} = 0. \tag{A.9}$$

When (A.7) holds, it follows from (A.5) and (A.6) that

$$P = \begin{pmatrix} 0 & 0 & A_1^T \\ 0 & P_{22} & A_2^T \\ P_{31} & 0 & P_{33} \end{pmatrix} \text{ and } B = \begin{pmatrix} B_{11} & 0 & P_{31}^{-1} \\ 0 & B_{22} & 0 \\ P_{31}^{-T} & 0 & 0 \end{pmatrix}, \tag{A.10}$$

where

$$P_{33} + P_{33}^T + P_{31}B_{11}P_{31}^T = -C. \tag{A.11}$$

In the caseof (A.8),

$$P = \begin{pmatrix} 0 & 0 & A_1^T \\ P_{21} & P_{22} & A_2^T \\ P_{31} & 0 & -C \end{pmatrix} \text{ and } B = \begin{pmatrix} B_{11} & 0 & P_{31}^{-1} \\ 0 & B_{22} & 0 \\ P_{31}^{-T} & 0 & 0 \end{pmatrix}, \tag{A.12}$$

as then

$$P_{33} + P_{31}B_{11}P_{31}^T = P_{33} - P_{33}B_{31}P_{31}^T = P_{33} - P_{33} = 0$$

from (A.5) and (A.8) and hence $P_{33} = P_{33}^T = -C$ from (A.6). Finally, (A.9) can only hold when $C = 0$, and is a special instance of (A.12).

## Case 3: (4.4a) and (4.5c) hold

If (4.4a) and (4.5c) hold, $P_{31}$, $P_{22}$, $B_{31}$ and $B_{22}$ are non singular, and

$$\begin{aligned} (M_G)_{31} &= P_{31}B_{31}^T A_1 + P_{33}B_{33}A_1, \\ (M_G)_{32} &= P_{31}B_{31}^T A_2 + P_{33}B_{33}A_2 + P_{33}(B_{31}P_{21}^T + B_{32}P_{22}^T) \\ \text{and } (M_G)_{33} &= P_{31}B_{31}^T P_{33}^T + P_{33}B_{33}P_{33}^T + P_{33}B_{31}P_{31}^T. \end{aligned}$$

Since $P_{31}$ and $B_{31}$ are non singular, requirement (4.3a) implies that either (A.5) holds and either $P_{33} = 0$ or $B_{33} = 0$, or

$$P_{33}B_{33} = I - P_{31}B_{31}^T \tag{A.13}$$

with nonzero $P_{33}$ and $B_{33}$. It is easy to see that it is not possible for requirement (4.3c) to hold when $P_{33} = 0$ unless $C = 0$, which leads to

$$P = \begin{pmatrix} 0 & 0 & A_1^T \\ P_{21}^T & P_{22} & A_2^T \\ B_{31}^{-T} & 0 & 0 \end{pmatrix} \text{ and } B = \begin{pmatrix} 0 & 0 & B_{31}^T \\ 0 & B_{22} & B_{32}^T \\ B_{31} & B_{32} & B_{33} \end{pmatrix} \tag{A.14}$$

in this special case. So suppose instead that (A.5) holds and that $B_{33} = 0$. In this case, the requirement (4.3c) is simply that

$$P_{33} + P_{33}^T = -C,$$

while (4.3b) additionally requires that

$$B_{31}P_{21}^T + B_{32}P_{22}^T = 0. \tag{A.15}$$

This results in

$$P = \begin{pmatrix} 0 & 0 & A_1^T \\ P_{21} & P_{22} & A_2^T \\ P_{31} & 0 & P_{33} \end{pmatrix} \text{ and } B = \begin{pmatrix} 0 & 0 & B_{31}^T \\ 0 & B_{22} & B_{32}^T \\ B_{31} & B_{32} & 0 \end{pmatrix}, \tag{A.16}$$

where

$$P_{21} = -P_{22}B_{32}^TB_{31}^{-T}, \quad P_{31} = B_{31}^{-T} \text{ and } P_{33} + P_{33}^T = -C. \tag{A.17}$$

Finally, suppose that (A.13) holds with nonzero $P_{33}$ and $B_{33}$. Then requirement (4.3c) is that

$$-C = P_{33}B_{33}P_{33}^T(I - P_{33}B_{33}^T)P_{33}^T + P_{33}(I - B_{33}P_{33}^T) = P_{33} + P_{33}^T - P_{33}B_{33}P_{33}^T$$

while once again (A.15) holds since $P_{22} \neq 0$. Thus, in this case

$$P = \begin{pmatrix} 0 & 0 & A_1^T \\ P_{21} & P_{22} & A_2^T \\ P_{31} & 0 & P_{33} \end{pmatrix} \text{ and } B = \begin{pmatrix} 0 & 0 & B_{31}^T \\ 0 & B_{22} & B_{32}^T \\ B_{31} & B_{32} & B_{33} \end{pmatrix}, \tag{A.18}$$

is possible provided that

$$B_{33} = P_{33}^{-1} + P_{33}^{-T} + P_{33}^{-1}CP_{33}^{-T}, \quad B_{31} = (I - B_{33}P_{33}^T)P_{31}^{-T} \text{ and } B_{32} = -B_{31}P_{21}^TP_{22}^{-T}. \tag{A.19}$$

## Case 4: (4.4b) and (4.5a) hold

If (4.4b) and (4.5a) hold, $P_{31}$, $P_{22}$, $B_{11}$, $B_{22}$ and $B_{33}$ are non singular, and

$$\begin{aligned} (M_G)_{31} &= P_{33}B_{33}A_1, \\ (M_G)_{32} &= P_{33}B_{33}A_2 + P_{32}B_{22}P_{22}^T \\ \text{and } (M_G)_{33} &= P_{33}B_{33}P_{33}^T + P_{32}B_{22}P_{32}^T + P_{31}B_{11}P_{31}^T. \end{aligned}$$

As in case 1, requirement (4.3a) implies that (A.1) holds (and thus that $P_{33}$ is symmetric). Requirement (4.3b) then forces $P_{32}B_{22}P_{22}^T = 0$, and thus that $P_{32} = 0$ since $P_{22}$ and $B_{22}$ are non singular. But then requirement (4.3c) leads once again to (A.2), and hence exactly the same conclusions as for case 1.

## Case 5: (4.4b) and (4.5b) hold

If (4.4b) and (4.5b) hold, $P_{31}$, $P_{22}$, $B_{31}$ and $B_{22}$ are non singular, and

$$\begin{aligned} (M_G)_{31} &= P_{31}B_{31}^TA_1, \\ (M_G)_{32} &= P_{31}B_{31}^TA_2 + (P_{31}B_{21}^T + P_{32}B_{22})P_{22}^T \\ \text{and } (M_G)_{33} &= P_{31}B_{31}^TP_{33}^T + (P_{31}B_{21}^T + P_{32}B_{22})P_{32}^T + P_{33}B_{31}P_{31}^T + (P_{31}B_{11} + P_{32}B_{21})P_{31}^T, \end{aligned}$$

As in case 2, requirement (4.3a) implies that (A.5) holds. Hence requirement (4.3b) and the non-singularity of $P_{22}$ together imply that

$$P_{31}B_{21}^T + P_{32}B_{22} = 0.$$

Thus either

$$B_{21} = 0 \text{ and } P_{32} = 0$$

or

$$P_{32} = -P_{31}B_{21}^TB_{22}^{-1} \text{ with nozero } B_{21} \text{ and } P_{32}$$

since $B_{31}$ and $P_{22}$ are non singular. The first of these two cases is identical to (A.10)–(A.11) under requirement (4.3c). Under the same requirement, simple manipulation for the second case gives

$$P = \begin{pmatrix} 0 & 0 & A_1^T \\ 0 & P_{22} & A_2^T \\ P_{31} & P_{32} & P_{33} \end{pmatrix} \text{ and } B = \begin{pmatrix} B_{11} & B_{21}^T & B_{31}^T \\ B_{21} & B_{22} & 0 \\ B_{31} & 0 & 0 \end{pmatrix}, \qquad \text{(A.20)}$$

where

$$P_{31} = B_{31}^{-T}, \ \ P_{32} = -P_{31}B_{21}^TB_{22}^{-1} \text{ and } P_{33} + P_{33}^T = -C - P_{31}(B_{11} - B_{21}^TB_{22}^{-1}B_{21})P_{31}^T. \qquad \text{(A.21)}$$

## Case 6: (4.4b) and (4.5c) hold

If (4.4b) and (4.5c) hold, $P_{31}$, $P_{22}$, $B_{31}$ and $B_{22}$ are non singular, and

$$\begin{aligned} (M_G)_{31} = &\ (P_{31}B_{31}^T + P_{32}B_{32}^T)A_1 + P_{33}B_{33}A_1, \\ (M_G)_{32} = &\ (P_{31}B_{31}^T + P_{32}B_{32}^T)A_2 + P_{33}B_{33}A_2 + P_{32}B_{22}P_{22}^T + P_{33}B_{32}P_{22}^T \\ \text{and}\ (M_G)_{33} = &\ (P_{31}B_{31}^T + P_{32}B_{32}^T)P_{33}^T + P_{33}B_{33}P_{33}^T + P_{32}B_{22}P_{32}^T + P_{33}(B_{31}P_{31}^T + B_{32}P_{32}^T). \end{aligned}$$

Requirement (4.3a) implies that either

$$P_{31}B_{31}^T + P_{32}B_{32}^T = I, \tag{A.22}$$

and either $P_{33} = 0$ or $B_{33} = 0$, or

$$P_{31}B_{31} = I - P_{33}B_{33} - P_{32}B_{32}^T \tag{A.23}$$

with nonzero $P_{33}$ and $B_{33}$. Just as in case 3, it is easy to see that it is not possible for requirement (4.3c) to hold when $P_{33} = 0$ unless $C = 0$, and in this case (A.14) holds with $P_{21} = 0$. So suppose instead that (A.22) holds and that $B_{33} = 0$. Then the non-singularity of $P_{22}$ and $B_{22}$ and requirement (4.3b) together imply that

$$P_{32} = -P_{33}B_{32}B_{22}^{-1}. \tag{A.24}$$

Finally, requirement (4.3b), (A.22) and (A.24) give that

$$P_{33} + P_{33}^T + P_{33}B_{32}B_{22}^{-1}B_{32}^T P_{33}^T = -C. \tag{A.25}$$

This results in

$$P = \begin{pmatrix} 0 & 0 & A_1^T \\ 0 & P_{22} & A_2^T \\ P_{31} & P_{32} & P_{33} \end{pmatrix} \ \text{and}\ B = \begin{pmatrix} 0 & 0 & B_{31}^T \\ 0 & B_{22} & B_{32}^T \\ B_{31} & B_{32} & 0 \end{pmatrix}, \tag{A.26}$$

where

$$\begin{aligned} &P_{33} + P_{33}^T + P_{33}B_{32}B_{22}^{-1}B_{32}^T P_{33}^T = -C, \\ &P_{32} = -P_{33}B_{32}B_{22}^{-1} \ \text{and}\ P_{31} = (I - P_{32}B_{32}^T)B_{31}^{-T}. \end{aligned} \tag{A.27}$$

Notice that although (A.25) restricts the choice of $B_{32}$ and $B_{22}$, it is easily satisfied, for example, when $B_{32} = 0$.

Finally, suppose that (A.23) holds with nonzero $P_{33}$ and $B_{33}$. Then once again the non-singularity of $P_{22}$ and $B_{22}$ and requirement (4.3b) together imply that (A.24) holds, while (A.23) and (A.24) show that requirement (4.3c) holds whenever

$$\begin{aligned} -C = &\ P_{33}B_{33}P_{33}^T + (I - P_{33}B_{33})P_{33}^T + P_{33}(I - B_{33}P_{33}^T) + P_{32}B_{22}P_{32}^T \\ = &\ P_{33} + P_{33}^T + P_{33}(B_{32}B_{22}^{-1}B_{32}^T - B_{33})P_{33}^T. \end{aligned}$$

This results in

$$P = \begin{pmatrix} 0 & 0 & A_1^T \\ 0 & P_{22} & A_2^T \\ P_{31} & P_{32} & P_{33} \end{pmatrix} \ \text{and}\ B = \begin{pmatrix} 0 & 0 & B_{31}^T \\ 0 & B_{22} & B_{32}^T \\ B_{31} & B_{32} & B_{33} \end{pmatrix}, \tag{A.28}$$

where

$$\begin{array}{l} P_{33} + P_{33}^T + P_{33}(B_{32}B_{22}^{-1}B_{32}^T - B_{33})P_{32}^T = -C, \\ P_{32} = -P_{33}B_{32}B_{22}^{-1} \text{ and } P_{31} = (I - P_{32}B_{32}^T - P_{33}B_{33})B_{31}^{-T}. \end{array} \tag{A.29}$$

Note that (A.26)–(A.27) is the special case $B_{33} = 0$ of (A.28)–(A.29).

## Case 7: (4.4c) and (4.5a) hold

If (4.4c) and (4.5a) hold, $P_{31}$, $P_{22}$, $B_{11}$, $B_{22}$ and $B_{33}$ are non singular, and

$$(M_G)_{31} = P_{31}B_{11}P_{11}^T, \;\; (M_G)_{32} = P_{31}B_{11}P_{21}^T \;\text{ and }\; (M_G)_{33} = P_{31}B_{11}P_{31}^T.$$

To satisfy requirements (4.3a)–(4.3c), $C$ must be non-singular and in this case

$$P_{11} = A_1^T, \;\; P_{21} = A_2^T, \;\; P_{31} = -C \;\text{ and }\; B_{11} = -C^{-1}.$$

This then leads to

$$P = \begin{pmatrix} A_1^T & 0 & A_1^T \\ A_2^T & P_{22} & A_2^T \\ C & 0 & 0 \end{pmatrix} \text{ and } B = \begin{pmatrix} -C^{-1} & 0 & 0 \\ 0 & B_{22} & 0 \\ 0 & 0 & B_{33} \end{pmatrix}. \tag{A.30}$$

## Case 8: (4.4c) and (4.5b) hold

If (4.4c) and (4.5b) hold, $P_{31}$, $P_{22}$, $B_{31}$ and $B_{22}$ are non singular, and

$$\begin{array}{rl} (M_G)_{31} = & P_{31}B_{11}P_{11}^T + P_{31}B_{31}^TA_1, \\ (M_G)_{32} = & P_{31}B_{11}P_{21}^T + P_{31}B_{21}^TP_{22}^T + P_{31}B_{31}^TA_2 \\ \text{and } (M_G)_{33} = & P_{31}B_{11}P_{31}^T. \end{array}$$

Requirements (4.3a)–(4.3c) (in reverse order) then imply that

$$\begin{array}{rl} B_{11} = & -P_{31}^{-1}CP_{31}^{-T}, \\ B_{31} = & P_{31}^{-T} - A_1^{-T}P_{11}B_{11} \\ \text{and } B_{21} = & P_{22}^{-1}(P_{21} - A_2^TA_1^{-T}P_{11})B_{11}. \end{array}$$

While there is very little reason to believe that $B_{31}$ will be easily invertible in general, it may be if $P_{11} = A_1^TM$ for some diagonal $M$ and if $P_{31}$ and $B_{11}$ are also diagonal. This then leads to

$$P = \begin{pmatrix} P_{11} & 0 & A_1^T \\ P_{21} & P_{22} & A_2^T \\ P_{31} & 0 & 0 \end{pmatrix} \text{ and } B = \begin{pmatrix} B_{11} & B_{21}^T & B_{31}^T \\ B_{21} & B_{22} & 0 \\ B_{31} & 0 & 0 \end{pmatrix}, \tag{A.31}$$

where

$$B_{11} = -P_{31}^{-1}CP_{31}^{-T}, \;\; B_{31} = P_{31}^{-T} - MB_{11}, \;\; B_{21} = P_{22}^{-1}(P_{21} - A_2^TM)B_{11} \;\text{ and }\; P_{11} = A_1^TM \tag{A.32}$$

for some suitable $M$.

## Case 9: (4.4c) and (4.5c) hold

If (4.4c) and (4.5c) hold, $P_{31}$, $P_{22}$, $B_{31}$ and $B_{22}$ are non singular, and

$$(M_G)_{31} = P_{31}B_{31}^T A_1, \;\; (M_G)_{32} = P_{31}B_{31}^T A_2 \;\text{ and }\; (M_G)_{33} = 0$$

We can only satisfy requirements (4.3a)–(4.3c) for this case if $C = 0$, and this gives

$$P = \begin{pmatrix} P_{11} & 0 & A_1^T \\ P_{21} & P_{22} & A_2^T \\ B_{31}^{-T} & 0 & 0 \end{pmatrix} \;\text{ and }\; B = \begin{pmatrix} 0 & 0 & B_{31}^T \\ 0 & B_{22} & B_{32}^T \\ B_{31} & B_{32} & B_{33} \end{pmatrix} \tag{A.33}$$

under such circumstances. Note that (A.14) is a special case of (A.33).

## Case 10: (4.4a) and (4.6) hold

If (4.4a) and (4.6) hold, $P_{31}$, $P_{22}$ and $B_{22}$ are non singular, and

$$\begin{aligned} (M_G)_{31} = &\; P_{31}B_{31}^T A_1 + P_{33}B_{33}A_1, \\ (M_G)_{32} = &\; P_{31}B_{31}^T A_2 + P_{33}B_{33}A_2 + P_{31}B_{11}P_{21}^T + P_{33}B_{31}P_{21}^T \\ \text{and } (M_G)_{33} = &\; P_{31}B_{31}^T P_{33}^T + P_{33}B_{33}P_{33}^T + P_{31}B_{11}P_{31}^T + P_{33}B_{31}P_{31}^T. \end{aligned}$$

To satisfy (4.3a) and (4.3b), necessarily

$$P_{31}B_{31}^T + P_{33}B_{33} = I \tag{A.34}$$

and either

$$P_{31}B_{11} + P_{33}B_{31} = 0. \tag{A.35}$$

or

$$P_{31}B_{11} + P_{33}B_{31} \neq 0 \;\text{ and }\; P_{21} = 0. \tag{A.36}$$

If (A.34) and (A.35) hold, requirement (4.3c) is simply that $P_{33} = -C$. If $C$ is invertible, this leads to

$$P = \begin{pmatrix} 0 & 0 & A_1^T \\ P_{21} & P_{22} & A_2^T \\ P_{31} & 0 & P_{33} \end{pmatrix} \;\text{ and }\; B = \begin{pmatrix} B_{11} & 0 & B_{31}^T \\ 0 & B_{22} & 0 \\ B_{31} & 0 & B_{33} \end{pmatrix}, \tag{A.37}$$

where

$$P_{33} = -C, \;\; P_{31}^T = -B_{11}^{-1}B_{31}^T P_{33}^T \;\text{ and }\; B_{33} = (I - B_{31}P_{31}^T)P_{33}^{-T}. \tag{A.38}$$

However, since solves with $B$ simply involve $B_{22}$ and

$$\begin{pmatrix} B_{11} & B_{31}^T \\ B_{31} & B_{33} \end{pmatrix} = \begin{pmatrix} B_{11} & 0 \\ B_{31} & I \end{pmatrix} \begin{pmatrix} B_{11}^{-1} & 0 \\ 0 & C^{-1} \end{pmatrix} \begin{pmatrix} B_{11}^T & B_{31}^T \\ 0 & I \end{pmatrix} \tag{A.39}$$

the block form of (A.39) indicate that only products with $C$, and not its inverse, are required when solving with $B$, and that $B_{33}$ need not be formed. If $C$ is singular (A.34) and (A.35) give that

$$P_{33} = -C, \;\; B_{31} = (I + B_{33}C)P_{31}^{-T} \;\text{ and }\; B_{11} = P_{31}^{-1}(C + CB_{33}C)P_{31}^{-T} \tag{A.40}$$

As before solves with $B$ simply involve $B_{22}$ and

$$\begin{pmatrix} B_{11} & B_{31}^T \\ B_{31} & B_{33} \end{pmatrix} = \begin{pmatrix} P_{31}^{-1} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} CB_{33}C + C & I + CB_{33} \\ I + B_{33}C & B_{33} \end{pmatrix} \begin{pmatrix} P_{31}^{-T} & 0 \\ 0 & I \end{pmatrix},$$

and thus we need to ensure that

$$\begin{pmatrix} CB_{33}C + C & I + CB_{33} \\ I + B_{33}C & B_{33} \end{pmatrix} = \begin{pmatrix} C & I \\ I & 0 \end{pmatrix} + \begin{pmatrix} C \\ I \end{pmatrix} B_{33} \begin{pmatrix} C & I \end{pmatrix} \tag{A.41}$$

is non-singular (and has the correct inertia). The possibility $B_{33} = 0$ is that given by (A.12) in Case 2, but an interesting alternative is when $B_{33}$ is chosen so that

$$B_{33}C = 0. \tag{A.42}$$

In this case, (A.40) becomes

$$P_{33} = -C, \;\; B_{31} = P_{31}^{-T} \;\text{ and }\; B_{11} = P_{31}^{-1}CP_{31}^{-T}, \tag{A.43}$$

and (A.41) gives

$$\begin{pmatrix} CB_{33}C + C & I + CB_{33} \\ I + B_{33}C & B_33 \end{pmatrix} = \begin{pmatrix} C & I \\ I & B_{33} \end{pmatrix} = \begin{pmatrix} I & 0 \\ B_{33} & I \end{pmatrix} \begin{pmatrix} C & I \\ I & 0 \end{pmatrix}$$

which is clearly (block) invertible.

If (A.34) and (A.36), requirement (4.3c) is that

$$-C = P_{33}^T + P_{31}B_{11}P_{21}^T + P_{33}B_{31}P_{21}^T,$$

and this leads to

$$P = \begin{pmatrix} 0 & 0 & A_1^T \\ 0 & P_{22} & A_2^T \\ P_{31} & 0 & P_{33} \end{pmatrix} \;\text{ and }\; B = \begin{pmatrix} B_{11} & 0 & B_{31}^T \\ 0 & B_{22} & 0 \\ B_{31} & 0 & B_{33} \end{pmatrix}, \tag{A.44}$$

where

$$P_{31} = (I - P_{33}B_{33})B_{31}^{-T} \;\text{ and }\; B_{11} = P_{31}^{-1}\left(P_{33}B_{33}P_{33}^T - C - P_{33} - P_{33}^T\right)P_{31}^{-T}. \tag{A.45}$$

A particularly simple case is when $P_{33} = 0$, for then

$$P = \begin{pmatrix} 0 & 0 & A_1^T \\ 0 & P_{22} & A_2^T \\ B_{31}^{-T} & 0 & 0 \end{pmatrix} \;\text{ and }\; B = \begin{pmatrix} B_{31}^TCB_{31} & 0 & B_{31}^T \\ 0 & B_{22} & 0 \\ B_{31} & 0 & B_{33} \end{pmatrix} \tag{A.46}$$

although even then $B$ will not normally be easily invertible unless $C$, $B_{31}$ and $B_{33}$ are diagonal.

## Case 11: (4.4b) and (4.6) hold

If (4.4b) and (4.6) hold, $P_{31}$, $P_{22}$ and $B_{22}$ are non singular, and

$$\begin{aligned} (M_G)_{31} = &\ P_{31}B_{31}^T A_1 + P_{33}B_{33}A_1, \\ (M_G)_{32} = &\ P_{31}B_{31}^T A_2 + P_{33}B_{33}A_2 + P_{32}B_{22}P_{22}^T \\ \text{and } (M_G)_{33} = &\ P_{31}B_{31}^T P_{33}^T + P_{33}B_{33}P_{33}^T + P_{33}B_{31}P_{31}^T + P_{32}B_{22}P_{32}^T + P_{31}B_{11}P_{31}^T. \end{aligned}$$

To satisfy requirement (4.3a), necessarily (A.34) holds. But then requirement (4.3b) and the non-singularity of $P_{22}$ and $B_{22}$ forces $P_{32} = 0$. This case is then simply a sub-case of the previous one.

## Case 12: (4.4c) and (4.6) hold

If (4.4c) and (4.6) hold, $P_{31}$, $P_{22}$ and $B_{22}$ are non singular, and

$$\begin{aligned} (M_G)_{31} = &\ P_{31}B_{11}P_{11}^T + P_{31}B_{31}^T A_1, \\ (M_G)_{32} = &\ P_{31}B_{11}P_{21}^T + P_{31}B_{31}^T A_2 \\ \text{and } (M_G)_{33} = &\ P_{31}B_{11}P_{31}^T. \end{aligned}$$

Just as for case 8, requirements (4.3a) and (4.3c) respctively imply that

$$B_{11} = -P_{31}^{-1}CP_{31}^{-T} \text{ and } B_{31} = P_{31}^{-T} - A_1^{-T}P_{11}B_{11}.$$

But requirement (4.3b) imposes that $B_{11}(P_{21}^T - P_{11}^T A_1^{-1}A_2) = 0$, which is certainly satisfied when

$$P_{21}^T = P_{11}^T A_1^{-1}A_2.$$

The latter is true, for example if

$$P_{11} = A_1^T M \text{ and } P_{21} = A_2^T M$$

for a given matrix $M$. In general, we thus have that

$$P = \begin{pmatrix} P_{11} & 0 & A_1^T \\ P_{21} & P_{22} & A_2^T \\ P_{31} & 0 & 0 \end{pmatrix} \text{ and } B = \begin{pmatrix} B_{11} & 0 & B_{31}^T \\ 0 & B_{22} & 0 \\ B_{31} & 0 & B_{33} \end{pmatrix} \tag{A.47}$$

where

$$B_{11} = -P_{31}^{-1}CP_{31}^{-T}, \; B_{31} = P_{31}^{-T} - A_1^{-T}P_{11}B_{11} \text{ and } P_{21} = A_2^T A_1^{-T}P_{11}.$$

The warnings concerning the easy invertibility of $B_{31}$ we mention for case 9 equally apply here, and so we actually require

$$B_{11} = -P_{31}^{-1}CP_{31}^{-T}, \; B_{31} = P_{31}^{-T} - MB_{11}, \; P_{11} = A_1^T M \text{ and } P_{21} = A_2^T M, \tag{A.48}$$

for some suitable (diagonal) $M$.

# Appendix B

Here we examine the matrix $G$ which arises for each of the families mentioned in Section 4.1. Note that for general $P$ and $B$ partioned as in (4.2), we have

$$\begin{aligned} G_{11} = {} & P_{11}B_{11}P_{11}^T + P_{11}B_{21}^T P_{12}^T + P_{11}B_{31}^T A_1 + P_{12}B_{21}P_{11}^T + P_{12}B_{22}P_{12}^T \\ & + P_{12}B_{32}^T A_1 + A_1^T B_{31}P_{11}^T + A_1^T B_{32}P_{12}^T + A_1^T B_{33}A_1 \\ G_{21} = {} & P_{21}B_{11}P_{11}^T + P_{21}B_{21}^T P_{12}^T + P_{21}B_{31}^T A_1 + P_{22}B_{21}P_{11}^T + P_{22}B_{22}P_{12}^T \\ & + P_{22}B_{32}^T A_1 + A_2^T B_{31}P_{11}^T + A_2^T B_{32}P_{12}^T + A_2^T B_{33}A_1 \\ \text{and } G_{22} = {} & P_{21}B_{11}P_{21}^T + P_{21}B_{21}^T P_{22}^T + P_{21}B_{31}^T A_2 + P_{22}B_{21}P_{21}^T + P_{22}B_{22}P_{22}^T \\ & + P_{22}B_{32}^T A_2 + A_2^T B_{31}P_{21}^T + A_2^T B_{32}P_{22}^T + A_2^T B_{33}A_2 \end{aligned}$$

## Family 1 (Appendix, A, Case 1): (4.4a) and (4.5a) hold

In this case

$$\begin{aligned} G_{11} = {} & A_1^T B_{33}A_1, \quad G_{21} = A_2^T B_{33}A_1 \\ \text{and } G_{22} = {} & P_{21}B_{11}P_{21}^T + P_{22}B_{22}P_{22}^T + A_2^T B_{33}A_2. \end{aligned}$$

Since $P_{21} = 0$ for Family 1, $G_{22}$ becomes

$$G_{22} = P_{22}B_{22}P_{22}^T + A_2^T B_{33}A_2.$$

## Families 2 and 3 (Appendix, A, Case 2): (4.4a) and (4.5b) hold

In this case

$$\begin{aligned} G_{11} = {} & 0, \\ G_{21} = {} & P_{21}B_{31}^T A_1 \\ \text{and } G_{22} = {} & P_{21}B_{11}P_{21}^T + P_{21}B_{21}^T P_{22}^T + P_{21}B_{31}^T A_2 + P_{22}B_{21}P_{21}^T + P_{22}B_{22}P_{22}^T \\ & + A_2^T B_{31}P_{21}^T. \end{aligned}$$

For Family 2, $B_{21} = 0$ and $P_{21} = 0$, so that $G_{21}$ and $G_{22}$ become

$$G_{21} = 0 \text{ and } G_{22} = P_{22}B_{22}P_{22}^T.$$

For Family 3, $B_{21} = 0$, so that $G_{22}$ becomes

$$G_{22} = P_{21}B_{11}P_{21}^T + P_{21}B_{31}^T A_2 + P_{22}B_{22}P_{22}^T + A_2^T B_{31}P_{21}^T.$$

## Families 4 and 5 (Appendix, A, Case 3): (4.4a) and (4.5c) hold

Here

$$\begin{aligned} G_{11} = {} & A_1^T B_{33}A_1, \\ G_{21} = {} & P_{21}B_{31}^T A_1 + P_{22}B_{32}^T A_1 + A_2^T B_{33}A_1 \\ \text{and } G_{22} = {} & P_{21}B_{31}^T A_2 + P_{22}B_{22}P_{22}^T + P_{22}B_{32}^T A_2 + A_2^T B_{31}P_{21}^T + A_2^T B_{32}P_{22}^T \\ & + A_2^T B_{33}A_2. \end{aligned}$$

For both families, (A.15) holds, and thus

$$G_{21} = A_2^T B_{33} A_1 \text{ and } G_{22} = P_{22} B_{22} P_{22}^T + A_2^T B_{33} A_2.$$

In addition, for Family 4, $B_{33} = 0$, and thus

$$G_{11} = 0, \quad G_{21} = 0 \text{ and } G_{22} = P_{22} B_{22} P_{22}^T.$$

## Family 6 (Appendix, A, Case 5): (4.4b) and (4.5b) hold

Here

$$G_{11} = 0, \quad G_{21} = 0 \text{ and } G_{22} = P_{22} B_{22} P_{22}^T.$$

## Family 7 (Appendix, A, Case 6): (4.4b) and (4.5c) hold

In this case

$$\begin{aligned} G_{11} &= A_1^T B_{33} A_1, \\ G_{21} &= P_{22} B_{32}^T A_1 + A_2^T B_{33} A_1 \\ \text{and } G_{22} &= P_{22} B_{22} P_{22}^T + P_{22} B_{32}^T A_2 + A_2^T B_{32} P_{22}^T + A_2^T B_{33} A_2. \end{aligned}$$

## Family 8 (Appendix, A, Case 7): (4.4c) and (4.5a) hold

Here

$$\begin{aligned} G_{11} &= P_{11} B_{11} P_{11}^T + A_1^T B_{33} A_1 \\ G_{21} &= P_{21} B_{11} P_{11}^T + A_2^T B_{33} A_1 \\ \text{and } G_{22} &= P_{21} B_{11} P_{21}^T + P_{22} B_{22} P_{22}^T + A_2^T B_{33} A_2. \end{aligned}$$

But since $P_{11} = A_1^T$, $P_{21} = A_2^T$ and $B_{11} = -C^{-1}$, we have

$$\begin{aligned} G_{11} &= A_1^T (B_{33} - C^{-1}) A_1 \\ G_{21} &= A_2^T (B_{33} - C^{-1}) A_1 \\ \text{and } G_{22} &= P_{22} B_{22} P_{22}^T + A_2^T (B_{33} - C^{-1}) A_2. \end{aligned}$$

## Family 9 (Appendix, A, Case 8): (4.4c) and (4.5b) hold

In this case

$$\begin{aligned} G_{11} &= P_{11} B_{11} P_{11}^T + P_{11} B_{31}^T A_1 + A_1^T B_{31} P_{11}^T, \\ G_{21} &= P_{21} B_{11} P_{11}^T + P_{21} B_{31}^T A_1 + P_{22} B_{21} P_{11}^T + A_2^T B_{31} P_{11}^T \\ \text{and } G_{22} &= P_{21} B_{11} P_{21}^T + P_{21} B_{21}^T P_{22}^T + P_{21} B_{31}^T A_2 + P_{22} B_{21} P_{21}^T + P_{22} B_{22} P_{22}^T \\ &\quad + A_2^T B_{31} P_{21}^T. \end{aligned}$$

### Family 10 (Appendix, A, Case 9): (4.4c) and (4.5c) hold

Here

$$\begin{aligned} G_{11} &= P_{11}B_{31}^T A_1 + A_1^T B_{31} P_{11}^T + A_1^T B_{33} A_1, \\ G_{21} &= P_{21}B_{31}^T A_1 + P_{22}B_{32}^T A_1 + A_2^T B_{31} P_{11}^T + A_2^T B_{33} A_1 \\ \text{and } G_{22} &= P_{21}B_{31}^T A_2 + P_{22}B_{22}P_{22}^T + P_{22}B_{32}^T A_2 + A_2^T B_{31} P_{21}^T + A_2^T B_{32} P_{22}^T \\ &\quad + A_2^T B_{33} A_2. \end{aligned}$$

### Families 11, 12 and 13 (Appendix, A, Case 10): (4.4a) and (4.6) hold

In this case

$$\begin{aligned} G_{11} &= A_1^T B_{33} A_1, \\ G_{21} &= P_{21}B_{31}^T A_1 + A_2^T B_{33} A_1 \\ \text{and } G_{22} &= P_{21}B_{11}P_{21}^T + P_{21}B_{31}^T A_2 + P_{22}B_{22}P_{22}^T + A_2^T B_{31} P_{21}^T + A_2^T B_{33} A_2. \end{aligned}$$

For Family 13, $P_{21} = 0$, and thus

$$\begin{aligned} G_{11} &= A_1^T B_{33} A_1, \\ G_{21} &= A_2^T B_{33} A_1 \\ \text{and } G_{22} &= P_{22}B_{22}P_{22}^T + A_2^T B_{33} A_2. \end{aligned}$$

### Family 14 (Appendix, A, Case 12): (4.4c) and (4.6) hold

Here

$$\begin{aligned} G_{11} &= P_{11}B_{11}P_{11}^T + P_{11}B_{31}^T A_1 + A_1^T B_{31} P_{11}^T + A_1^T B_{33} A_1, \\ G_{21} &= P_{21}B_{11}P_{11}^T + P_{21}B_{31}^T A_1 + A_2^T B_{31} P_{11}^T + A_2^T B_{33} A_1 \\ \text{and } G_{22} &= P_{21}B_{11}P_{21}^T + P_{21}B_{31}^T A_2 + P_{22}B_{22}P_{22}^T + A_2^T B_{31} P_{21}^T + A_2^T B_{33} A_2 \end{aligned}$$

# Appendix C

Here we give the raw data for each of the experiments reported in Section 5. For each algorithm used, we report the CPU time (in seconds) needed to construct the preconditioner, along with the number of PPCG iterations and total time (including the construction cost) required to solve the problem. Both low- and high(er)-accuracy solutions are recorded for both of the examples of $C$ we considered.

Table C.1: CUTEr QP problems—residual decrease of at least $10^{-2}$ and $C = I$

| | Explicit factors | | | | | | Implicit factors | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $G = H$ | | | $G = I$ | | | Family 1 | | | Family 2(a) | | | Family 2(b) | | |
| name | fact. | iter. | total | fact. | iter. | total | fact. | iter. | total | fact. | iter. | total | fact. | iter. | total |
| AUG2DCQP | 0.38 | 1 | 0.45 | 0.12 | 1 | 0.18 | 0.07 | 13 | 0.19 | 0.07 | 267 | 1.94 | 0.02 | 36 | 0.30 |
| AUG2DQP | 0.12 | 1 | 0.18 | 0.12 | 1 | 0.18 | 0.06 | 13 | 0.17 | 0.03 | 268 | 1.89 | 0.02 | 36 | 0.29 |
| AUG3DCQP | 0.16 | 1 | 0.24 | 0.16 | 1 | 0.24 | 0.06 | 24 | 0.24 | 0.03 | 96 | 0.74 | 0.03 | 37 | 0.33 |
| AUG3DQP | 0.16 | 1 | 0.24 | 0.16 | 1 | 0.24 | 0.07 | 25 | 0.26 | 0.03 | 96 | 0.75 | 0.03 | 37 | 0.33 |
| BLOCKQP1 | 5.03 | 1 | 33.28 | 4.98 | 1 | 33.15 | 0.14 | 1 | 28.20 | 0.06 | 1 | 28.18 | 0.06 | 1 | 28.09 |
| BLOCKQP2 | 4.98 | 1 | 33.13 | 5.02 | 1 | 33.14 | 0.14 | 1 | 28.22 | 0.06 | 1 | 28.17 | 0.07 | 1 | 28.16 |
| BLOCKQP3 | 4.96 | 1 | 33.03 | 5.04 | 1 | 33.13 | 0.14 | 1 | 28.08 | 0.06 | 1 | 28.10 | 0.06 | 1 | 28.07 |
| BLOWEYA | 0.27 | 1 | 0.33 | 16.58 | 1 | 16.69 | 0.06 | 1 | 0.10 | 0.05 | 1 | 0.09 | 0.05 | 1 | 0.09 |
| BLOWEYB | 0.28 | 1 | 0.34 | 16.52 | 1 | 16.63 | 0.06 | 1 | 0.10 | 0.05 | 1 | 0.09 | 0.05 | 1 | 0.08 |
| BLOWEYC | 0.29 | 1 | 0.35 | 14.75 | 1 | 14.87 | 0.07 | 1 | 0.11 | 0.05 | 1 | 0.09 | 0.05 | 1 | 0.09 |
| CONT-050 | 0.36 | 1 | 0.53 | 0.61 | 1 | 0.81 | 0.05 | 1 | 0.17 | 0.01 | 1 | 0.13 | 0.01 | 1 | 0.13 |
| CONT-101 | 2.17 | 1 | 3.33 | 4.39 | 1 | 5.84 | 0.20 | 0 | 1.02 | 0.06 | 0 | 0.87 | 0.05 | 0 | 0.87 |
| CONT-201 | 13.18 | 1 | 21.04 | 29.63 | 1 | 38.82 | 0.86 | 0 | 6.94 | 0.24 | 0 | 6.27 | 0.24 | 0 | 6.30 |
| CONT1-10 | 2.16 | 1 | 3.35 | 4.22 | 1 | 5.57 | 0.20 | 1 | 1.08 | 0.05 | 1 | 0.93 | 0.05 | 1 | 0.92 |
| CONT1-20 | 13.33 | 1 | 21.64 | 29.07 | 1 | 38.37 | 0.85 | 0 | 7.48 | 0.25 | 0 | 6.84 | 0.25 | 0 | 6.78 |
| CONT5-QP | ran out of memory | | | ran out of memory | | | 0.91 | 1 | 6.63 | 0.24 | 1 | 5.94 | 0.25 | 1 | 5.92 |
| CVXQP1 | 48.16 | 1 | 50.38 | 139.83 | 1 | 142.34 | 0.18 | 2 | 0.36 | 0.13 | 1310 | 43.43 | 0.12 | 1258 | 42.24 |
| CVXQP2 | 27.27 | 1 | 28.66 | 29.21 | 1 | 30.67 | 0.11 | 3 | 0.21 | 0.30 | 523 | 21.98 | 0.30 | 130 | 5.63 |
| CVXQP3 | 53.57 | 1 | 56.33 | 78.67 | 1 | 82.48 | 0.45 | 1 | 0.89 | 0.16 | 1 | 0.59 | 0.06 | 1 | 0.48 |
| DUALC1 | 0.13 | 1 | 0.23 | 0.03 | 1 | 0.06 | 0.01 | 1 | 0.03 | 0.01 | 1 | 0.02 | 0.01 | 3 | 0.03 |
| DUALC2 | 0.03 | 1 | 0.05 | 0.03 | 1 | 0.05 | 0.01 | 1 | 0.04 | 0.01 | 1 | 0.03 | 0.01 | 0 | 0.02 |
| DUALC5 | 0.04 | 1 | 0.07 | 0.13 | 1 | 0.16 | 0.02 | 10 | 0.06 | 0.01 | 1 | 0.03 | 0.01 | 563 | 0.81 |
| DUALC8 | 0.10 | 1 | 0.23 | 0.10 | 1 | 0.20 | 0.04 | 1 | 0.15 | 0.01 | 1 | 0.11 | 0.01 | 7 | 0.12 |
| GOULDQP2 | 0.85 | 1 | 1.04 | 0.79 | 1 | 1.04 | 0.27 | 4 | 0.45 | 0.12 | 13 | 0.60 | 0.11 | 14 | 0.65 |
| GOULDQP3 | 0.81 | 1 | 0.99 | 0.92 | 1 | 1.23 | 0.27 | 4 | 0.46 | 0.11 | 13 | 0.78 | 0.11 | 15 | 0.98 |
| KSIP | 0.50 | 1 | 1.04 | 0.50 | 1 | 1.02 | 0.06 | 2 | 0.59 | 0.01 | 2 | 0.55 | 0.01 | 2 | 0.55 |
| MOSARQP1 | 0.09 | 1 | 0.13 | 0.09 | 1 | 0.13 | 0.04 | 7 | 0.09 | 0.02 | 5 | 0.08 | 0.02 | 5 | 0.08 |
| NCVXQP1 | 117.67 | 1 | 119.93 | 141.30 | 1 | 143.77 | 0.14 | 2 | 0.32 | 9.66 | 1420 | 69.96 | 9.64 | 676 | 38.51 |
| NCVXQP2 | 129.22 | 1 | 131.48 | 141.86 | 1 | 144.34 | 0.14 | 3 | 0.35 | 9.64 | 1 | 9.82 | 9.65 | 31 | 11.12 |
| NCVXQP3 | 128.41 | 1 | 130.66 | 129.05 | 1 | 131.54 | 0.14 | 2 | 0.32 | 9.64 | 1197 | 60.17 | 8.36 | 1047 | 52.77 |
| NCVXQP4 | 89.54 | 1 | 90.99 | 93.00 | 1 | 94.49 | 0.12 | 2 | 0.20 | 19.55 | 564 | 51.00 | 19.44 | 4 | 19.73 |
| NCVXQP5 | 89.18 | 1 | 90.61 | 84.34 | 1 | 85.80 | 0.14 | 3 | 0.25 | 19.56 | 790 | 62.90 | 22.38 | 9 | 22.94 |
| NCVXQP6 | 88.20 | 1 | 89.58 | 91.74 | 1 | 93.18 | 0.14 | 3 | 0.24 | 18.91 | 522 | 47.40 | 18.91 | 161 | 27.79 |
| NCVXQP7 | 628.16 | 1 | 632.70 | 82.54 | 1 | 86.56 | 0.34 | 1 | 0.78 | 0.17 | 1 | 0.61 | 0.06 | 1 | 0.47 |
| NCVXQP8 | 61.02 | 1 | 64.21 | 84.22 | 1 | 88.60 | 0.28 | 1 | 0.73 | 0.19 | 1 | 0.65 | 0.07 | 1 | 0.52 |
| NCVXQP9 | 54.51 | 1 | 57.44 | 85.05 | 1 | 88.96 | 0.29 | 1 | 0.74 | 2.20 | 1 | 2.66 | 2.10 | 1 | 2.53 |
| POWELL20 | 0.34 | 1 | 0.47 | 0.32 | 1 | 0.46 | 0.14 | 1 | 0.20 | 0.06 | 13 | 0.31 | 0.06 | 14 | 0.32 |
| PRIMAL1 | 0.11 | 1 | 0.76 | 0.11 | 1 | 0.12 | 0.05 | 19 | 0.08 | 0.01 | 8 | 0.03 | 0.01 | 2 | 0.02 |
| PRIMAL2 | 0.16 | 1 | 0.17 | 0.16 | 1 | 0.17 | 0.05 | 18 | 0.09 | 0.02 | 2 | 0.03 | 0.03 | 2 | 0.04 |
| PRIMAL3 | 0.57 | 1 | 0.60 | 0.59 | 1 | 0.62 | 0.04 | 25 | 0.14 | 0.02 | 2 | 0.04 | 0.01 | 2 | 0.03 |
| PRIMAL4 | 0.30 | 1 | 0.32 | 0.30 | 1 | 0.32 | 0.05 | 15 | 0.11 | 0.01 | 2 | 0.03 | 0.02 | 2 | 0.03 |
| PRIMALC1 | 0.01 | 1 | 0.02 | 0.02 | 1 | 0.02 | 0.02 | 2 | 0.03 | 0.01 | 248 | 0.25 | 0.01 | 30 | 0.04 |
| PRIMALC2 | 0.02 | 1 | 0.02 | 0.03 | 1 | 0.03 | 0.03 | 2 | 0.03 | 0.01 | 245 | 0.24 | 0.01 | 245 | 0.24 |
| PRIMALC5 | 0.02 | 1 | 0.02 | 0.02 | 1 | 0.02 | 0.05 | 2 | 0.05 | 0.02 | 7 | 0.03 | 0.02 | 7 | 0.03 |
| PRIMALC8 | 0.04 | 1 | 0.04 | 0.03 | 1 | 0.04 | 0.02 | 2 | 0.03 | 0.02 | 64 | 0.12 | 0.01 | 6 | 0.02 |
| QPBAND | 0.42 | 1 | 0.55 | 0.42 | 1 | 0.57 | 0.19 | 2 | 0.26 | 0.15 | 16 | 0.77 | 0.15 | 14 | 0.71 |
| QPNBAND | 0.53 | 1 | 0.70 | 0.43 | 1 | 0.59 | 0.18 | 3 | 0.27 | 0.15 | 12 | 0.61 | 0.15 | 12 | 0.63 |
| QPCBOEI1 | 0.05 | 1 | 0.09 | 0.06 | 1 | 0.10 | 0.02 | 16 | 0.09 | 0.01 | 2 | 0.04 | 0.01 | 2 | 0.04 |
| QPCBOEI2 | 0.09 | 1 | 0.11 | 0.09 | 1 | 0.11 | 0.02 | 2 | 0.03 | 0.02 | 1 | 0.02 | 0.02 | 1 | 0.03 |
| QPNBOEI1 | 0.43 | 1 | 0.47 | 0.05 | 1 | 0.09 | 0.03 | 16 | 0.09 | 0.01 | 3 | 0.05 | 0.01 | 2 | 0.05 |
| QPNBOEI2 | 0.11 | 1 | 0.12 | 0.09 | 1 | 0.11 | 0.02 | 2 | 0.03 | 0.01 | 1 | 0.02 | 0.01 | 1 | 0.02 |
| QPCSTAIR | 0.05 | 1 | 0.11 | 0.05 | 1 | 0.11 | 0.02 | 3 | 0.08 | 0.01 | 13 | 0.09 | 0.01 | 5 | 0.07 |
| QPNSTAIR | 0.05 | 1 | 0.11 | 0.06 | 1 | 0.12 | 0.02 | 4 | 0.08 | 0.01 | 9 | 0.08 | 0.01 | 5 | 0.07 |
| SOSQP1 | 0.12 | 1 | 0.17 | 0.12 | 1 | 0.18 | 0.07 | 1 | 0.10 | 0.03 | 1 | 0.06 | 0.03 | 1 | 0.06 |
| STCQP2 | 0.86 | 1 | 0.96 | 1.47 | 1 | 1.62 | 0.05 | 4 | 0.12 | 0.09 | 1 | 0.13 | 0.09 | 2622 | 38.05 |
| STNQP2 | 63.37 | 1 | 63.72 | 66.32 | 1 | 66.82 | 0.12 | 8 | 0.38 | 6.52 | 247 | 17.24 | 0.27 | 15 | 0.86 |
| UBH1 | 0.34 | 1 | 0.52 | 0.33 | 1 | 0.52 | 0.13 | 2 | 0.29 | 0.05 | 1 | 0.17 | 0.05 | 4 | 0.23 |

Table C.2: CUTEr QP problems—residual decrease of at least $10^{-8}$ and $C = I$

| | Explicit factors | | | | | | Implicit factors | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $G = H$ | | | $G = I$ | | | Family 1 | | | Family 2(a) | | | Family 2(b) | | |
| name | fact. | iter. | total | fact. | iter. | total | fact. | iter. | total | fact. | iter. | total | fact. | iter. | total |
| AUG2DCQP | 0.38 | 1 | 0.44 | 0.12 | 1 | 0.18 | 0.07 | 157 | 1.11 | 0.07 | 1220 | 8.48 | 0.02 | 89 | 0.66 |
| AUG2DQP | 0.12 | 1 | 0.18 | 0.12 | 1 | 0.18 | 0.06 | 165 | 1.14 | 0.03 | 1271 | 8.83 | 0.02 | 670 | 4.60 |
| AUG3DCQP | 0.16 | 1 | 0.24 | 0.16 | 1 | 0.24 | 0.06 | 106 | 0.75 | 0.03 | 1684 | 11.91 | 0.03 | 2621 | 18.50 |
| AUG3DQP | 0.16 | 1 | 0.24 | 0.16 | 1 | 0.24 | 0.07 | 106 | 0.76 | 0.03 | 273 | 1.98 | 0.03 | 50 | 0.42 |
| BLOCKQP1 | 5.03 | 1 | 33.06 | 4.98 | 1 | 33.11 | 0.14 | 1 | 28.22 | 0.06 | 2 | 28.20 | 0.06 | 2 | 28.14 |
| BLOCKQP2 | 4.98 | 1 | 33.02 | 5.02 | 1 | 33.31 | 0.14 | 1 | 28.27 | 0.06 | 2 | 28.15 | 0.07 | 2 | 28.13 |
| BLOCKQP3 | 4.96 | 1 | 33.06 | 5.04 | 1 | 33.15 | 0.14 | 1 | 28.24 | 0.06 | 2 | 28.17 | 0.06 | 2 | 28.23 |
| BLOWEYA | 0.27 | 1 | 0.32 | 16.58 | 1 | 16.69 | 0.06 | 1 | 0.09 | 0.05 | 1 | 0.09 | 0.05 | 1 | 0.09 |
| BLOWEYB | 0.28 | 1 | 0.33 | 16.52 | 1 | 16.63 | 0.06 | 1 | 0.10 | 0.05 | 1 | 0.09 | 0.05 | 1 | 0.08 |
| BLOWEYC | 0.29 | 1 | 0.35 | 14.75 | 1 | 14.87 | 0.07 | 1 | 0.11 | 0.05 | 1 | 0.09 | 0.05 | 1 | 0.09 |
| CONT-050 | 0.36 | 1 | 0.53 | 0.61 | 1 | 0.81 | 0.05 | 9 | 0.23 | 0.01 | 28 | 0.34 | 0.01 | 29 | 0.35 |
| CONT-101 | 2.17 | 1 | 3.33 | 4.39 | 1 | 5.82 | 0.20 | 0 | 1.03 | 0.06 | 0 | 0.86 | 0.05 | 0 | 0.84 |
| CONT-201 | 13.18 | 1 | 21.05 | 29.63 | 1 | 38.79 | 0.86 | 0 | 6.97 | 0.24 | 0 | 6.25 | 0.24 | 0 | 6.34 |
| CONT1-10 | 2.16 | 1 | 3.34 | 4.22 | 1 | 5.57 | 0.20 | 6 | 1.26 | 0.05 | 28 | 1.80 | 0.05 | 30 | 1.90 |
| CONT1-20 | 13.33 | 1 | 21.66 | 29.07 | 1 | 38.44 | 0.85 | 0 | 7.52 | 0.25 | 0 | 6.76 | 0.25 | 0 | 6.77 |
| CONT5-QP | ran out of memory | | | ran out of memory | | | 0.91 | 1 | 6.71 | 0.24 | 110 | 30.72 | 0.25 | 147 | 37.19 |
| CVXQP1 | 48.16 | 1 | 50.39 | 139.83 | 1 | 142.36 | 0.18 | 51 | 1.56 | 0.13 | 9237 | 305.76 | 0.12 | 4165 | 138.96 |
| CVXQP2 | 27.27 | 1 | 28.67 | 29.21 | 1 | 30.67 | 0.11 | 369 | 7.39 | 0.30 | 9400 | 380.46 | 0.30 | 1353 | 55.55 |
| CVXQP3 | 53.57 | 1 | 56.32 | 78.67 | 1 | 82.50 | 0.45 | 3 | 0.98 | 0.16 | 9291 | 369.75 | 0.06 | 5782 | 224.38 |
| DUALC1 | 0.13 | 1 | 0.14 | 0.03 | 1 | 0.05 | 0.01 | 16 | 0.05 | 0.01 | 2 | 0.02 | 0.01 | 15 | 0.04 |
| DUALC2 | 0.03 | 1 | 0.05 | 0.03 | 1 | 0.05 | 0.01 | 9 | 0.05 | 0.01 | 1 | 0.03 | 0.01 | 0 | 0.02 |
| DUALC5 | 0.04 | 1 | 0.07 | 0.13 | 1 | 0.16 | 0.02 | 11 | 0.06 | 0.01 | 135 | 0.22 | 0.01 | 563 | 0.82 |
| DUALC8 | 0.10 | 1 | 0.20 | 0.10 | 1 | 0.21 | 0.04 | 13 | 0.18 | 0.01 | 997 | 2.16 | 0.01 | 14 | 0.13 |
| GOULDQP2 | 0.85 | 1 | 1.04 | 0.79 | 1 | 1.04 | 0.27 | 17 | 0.87 | 0.12 | 317 | 11.63 | 0.11 | 715 | 25.62 |
| GOULDQP3 | 0.81 | 1 | 0.99 | 0.92 | 1 | 1.23 | 0.27 | 18 | 0.93 | 0.11 | 96 | 4.63 | 0.11 | 673 | 31.52 |
| KSIP | 0.50 | 1 | 1.02 | 0.50 | 1 | 1.03 | 0.06 | 8 | 0.61 | 0.01 | 6 | 0.57 | 0.01 | 6 | 0.57 |
| MOSARQP1 | 0.09 | 1 | 0.13 | 0.09 | 1 | 0.13 | 0.04 | 50 | 0.27 | 0.02 | 295 | 2.07 | 0.02 | 952 | 6.58 |
| NCVXQP1 | 117.67 | 1 | 119.93 | 141.30 | 1 | 143.77 | 0.14 | 61 | 1.73 | 9.66 | 9557 | 416.62 | 9.64 | 1802 | 86.77 |
| NCVXQP2 | 129.22 | 1 | 131.49 | 141.86 | 1 | 144.36 | 0.14 | 9942 | 236.88 | 9.64 | 1 | 9.82 | 9.65 | 31 | 11.10 |
| NCVXQP3 | 128.41 | 1 | 130.67 | 129.05 | 1 | 131.54 | 0.14 | 62 | 1.78 | 9.64 | 9877 | 443.35 | 8.36 | 2618 | 119.47 |
| NCVXQP4 | 89.54 | 1 | 90.98 | 93.00 | 1 | 94.49 | 0.12 | 8373 | 163.19 | 19.55 | 7877 | 438.09 | 19.44 | 416 | 42.25 |
| NCVXQP5 | 89.18 | 1 | 90.59 | 84.34 | 1 | 85.79 | 0.14 | 8263 | 160.94 | 19.56 | 1069 | 78.17 | 22.38 | 61 | 25.79 |
| NCVXQP6 | 88.20 | 1 | 89.60 | 91.74 | 1 | 93.18 | 0.14 | 9041 | 175.54 | 18.91 | 9394 | 531.64 | 18.91 | 248 | 32.56 |
| NCVXQP7 | 628.16 | 1 | 632.76 | 82.54 | 1 | 86.49 | 0.34 | 3 | 0.84 | 0.17 | 9736 | 373.63 | 0.06 | 4650 | 180.96 |
| NCVXQP8 | 61.02 | 1 | 64.21 | 84.22 | 1 | 88.69 | 0.28 | 3 | 0.81 | 0.19 | 9994 | 426.71 | 0.07 | 5460 | 215.20 |
| NCVXQP9 | 54.51 | 1 | 57.41 | 85.05 | 1 | 89.06 | 0.29 | 3 | 0.79 | 2.20 | 1790 | 81.11 | 2.10 | 1124 | 50.49 |
| POWELL20 | 0.34 | 1 | 0.47 | 0.32 | 1 | 0.46 | 0.14 | 1 | 0.20 | 0.06 | 3581 | 54.93 | 0.06 | 82 | 1.37 |
| PRIMAL1 | 0.11 | 1 | 0.12 | 0.11 | 1 | 0.12 | 0.05 | 172 | 0.30 | 0.01 | 21 | 0.05 | 0.01 | 31 | 0.07 |
| PRIMAL2 | 0.16 | 1 | 0.17 | 0.16 | 1 | 0.17 | 0.05 | 132 | 0.31 | 0.02 | 23 | 0.08 | 0.03 | 37 | 0.12 |
| PRIMAL3 | 0.57 | 1 | 0.60 | 0.59 | 1 | 0.62 | 0.04 | 117 | 0.44 | 0.02 | 36 | 0.16 | 0.01 | 37 | 0.16 |
| PRIMAL4 | 0.30 | 1 | 0.32 | 0.30 | 1 | 0.32 | 0.05 | 62 | 0.27 | 0.01 | 13 | 0.08 | 0.02 | 53 | 0.23 |
| PRIMALC1 | 0.01 | 1 | 0.02 | 0.02 | 1 | 0.02 | 0.02 | 6 | 0.03 | 0.01 | 248 | 0.25 | 0.01 | 132 | 0.14 |
| PRIMALC2 | 0.02 | 34 | 0.05 | 0.03 | 1 | 0.03 | 0.03 | 4 | 0.03 | 0.01 | 245 | 0.23 | 0.01 | 245 | 0.24 |
| PRIMALC5 | 0.02 | 1 | 0.02 | 0.02 | 1 | 0.02 | 0.05 | 5 | 0.05 | 0.02 | 16 | 0.04 | 0.02 | 14 | 0.03 |
| PRIMALC8 | 0.04 | 1 | 0.04 | 0.03 | 1 | 0.04 | 0.02 | 5 | 0.03 | 0.02 | 536 | 0.81 | 0.01 | 46 | 0.08 |
| QPBAND | 0.42 | 1 | 0.55 | 0.42 | 1 | 0.57 | 0.19 | 7 | 0.37 | 0.15 | 50 | 1.97 | 0.15 | 224 | 8.36 |
| QPNBAND | 0.53 | 1 | 0.70 | 0.43 | 1 | 0.58 | 0.18 | 7 | 0.36 | 0.15 | 30 | 1.24 | 0.15 | 159 | 5.98 |
| QPCBOEI1 | 0.05 | 1 | 0.09 | 0.06 | 1 | 0.10 | 0.02 | 113 | 0.28 | 0.01 | 222 | 0.51 | 0.01 | 23 | 0.09 |
| QPCBOEI2 | 0.09 | 1 | 0.11 | 0.09 | 1 | 0.11 | 0.02 | 4 | 0.03 | 0.02 | 2 | 0.03 | 0.02 | 2 | 0.03 |
| QPNBOEI1 | 0.43 | 1 | 0.47 | 0.05 | 1 | 0.09 | 0.03 | 114 | 0.29 | 0.01 | 20 | 0.08 | 0.01 | 24 | 0.09 |
| QPNBOEI2 | 0.11 | 1 | 0.12 | 0.09 | 1 | 0.11 | 0.02 | 4 | 0.03 | 0.01 | 2 | 0.02 | 0.01 | 2 | 0.02 |
| QPCSTAIR | 0.05 | 1 | 0.11 | 0.05 | 1 | 0.11 | 0.02 | 144 | 0.35 | 0.01 | 142 | 0.35 | 0.01 | 38 | 0.14 |
| QPNSTAIR | 0.05 | 1 | 0.11 | 0.06 | 1 | 0.12 | 0.02 | 145 | 0.35 | 0.01 | 135 | 0.34 | 0.01 | 28 | 0.12 |
| SOSQP1 | 0.12 | 1 | 0.17 | 0.12 | 1 | 0.18 | 0.07 | 3 | 0.13 | 0.03 | 18 | 0.27 | 0.03 | 34 | 0.46 |
| STCQP2 | 0.86 | 1 | 0.96 | 1.47 | 1 | 1.62 | 0.05 | 92 | 1.12 | 0.09 | 1 | 0.13 | 0.09 | 6140 | 89.14 |
| STNQP2 | 63.37 | 1 | 63.72 | 66.32 | 1 | 66.82 | 0.12 | 5141 | 129.65 | 6.52 | 4747 | 177.13 | 0.27 | 5966 | 207.81 |
| UBH1 | 0.34 | 1 | 0.53 | 0.33 | 1 | 0.52 | 0.13 | 30 | 0.87 | 0.05 | 472 | 10.12 | 0.05 | 47 | 1.13 |

Table C.3: CUTEr QP problems—residual decrease of at least $10^{-2}$ and $C$ given by (5.9)

| | Explicit factors | | | | | | Implicit factors | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $G = H$ | | | $G = I$ | | | Family 1 | | | Family 2(a) | | | Family 2(b) | | |
| name | fact. | iter. | total | fact. | iter. | total | fact. | iter. | total | fact. | iter. | total | fact. | iter. | total |
| AUG2DCQP | 0.13 | 1 | 0.19 | 0.13 | 1 | 0.19 | 0.06 | 19 | 0.21 | 0.02 | 43 | 0.35 | 0.02 | 1 | 0.06 |
| AUG2DQP | 0.12 | 1 | 0.18 | 0.12 | 1 | 0.18 | 0.06 | 19 | 0.21 | 0.02 | 43 | 0.35 | 0.02 | 1 | 0.06 |
| AUG3DCQP | 0.16 | 2 | 0.25 | 0.17 | 2 | 0.25 | 0.06 | 14 | 0.19 | 0.03 | 12 | 0.15 | 0.03 | 10 | 0.14 |
| AUG3DQP | 0.17 | 2 | 0.25 | 0.17 | 2 | 0.26 | 0.06 | 14 | 0.18 | 0.03 | 13 | 0.16 | 0.03 | 10 | 0.14 |
| BLOCKQP1 | 4.94 | 2 | 32.95 | 4.93 | 2 | 33.05 | 0.14 | 1 | 28.14 | 0.06 | 1 | 28.07 | 0.07 | 1 | 28.10 |
| BLOCKQP2 | 4.93 | 2 | 32.84 | 4.87 | 2 | 33.00 | 0.14 | 1 | 28.11 | 0.07 | 1 | 28.11 | 0.07 | 1 | 27.99 |
| BLOCKQP3 | 4.94 | 2 | 32.87 | 4.81 | 2 | 32.99 | 0.14 | 1 | 28.20 | 0.06 | 1 | 28.11 | 0.06 | 1 | 28.02 |
| BLOWEYA | 0.25 | 1 | 0.31 | 14.64 | 1 | 14.73 | 0.06 | 1 | 0.09 | 0.02 | 1 | 0.06 | 0.02 | 1 | 0.06 |
| BLOWEYB | 0.26 | 1 | 0.31 | 14.72 | 1 | 14.82 | 0.06 | 1 | 0.09 | 0.03 | 1 | 0.06 | 0.02 | 1 | 0.06 |
| BLOWEYC | 0.26 | 1 | 0.31 | 12.99 | 1 | 13.09 | 0.06 | 1 | 0.10 | 0.03 | 1 | 0.07 | 0.03 | 1 | 0.06 |
| CONT-050 | 0.36 | 1 | 0.53 | 0.60 | 1 | 0.79 | 0.05 | 1 | 0.17 | 0.01 | 1 | 0.13 | 0.01 | 1 | 0.13 |
| CONT-101 | 2.17 | 1 | 3.33 | 4.33 | 1 | 5.72 | 0.20 | 0 | 1.03 | 0.05 | 0 | 0.86 | 0.05 | 0 | 0.84 |
| CONT-201 | 13.18 | 1 | 20.95 | 29.39 | 1 | 38.16 | 0.87 | 0 | 6.95 | 0.23 | 0 | 6.46 | 0.23 | 0 | 6.25 |
| CONT1-10 | 2.16 | 1 | 3.34 | 4.14 | 1 | 5.47 | 0.20 | 1 | 1.11 | 0.05 | 1 | 0.93 | 0.05 | 1 | 0.93 |
| CONT1-20 | 13.24 | 1 | 21.52 | 28.77 | 1 | 38.13 | 0.88 | 0 | 7.48 | 0.22 | 0 | 6.80 | 0.22 | 0 | 6.72 |
| CONT5-QP | ran out of memory | | | ran out of memory | | | 0.87 | 1 | 6.50 | 0.25 | 1 | 5.87 | 0.25 | 1 | 5.83 |
| CVXQP1 | 0.48 | 2 | 0.75 | 0.46 | 2 | 0.79 | 0.14 | 2 | 0.33 | 0.06 | 1 | 0.22 | 0.06 | 1 | 0.22 |
| CVXQP2 | 0.16 | 2 | 0.29 | 0.16 | 2 | 0.29 | 0.11 | 2 | 0.20 | 0.07 | 1 | 0.13 | 0.07 | 1 | 0.13 |
| CVXQP3 | 0.71 | 2 | 1.29 | 505.39 | 2 | 506.87 | 0.16 | 1 | 0.59 | 0.05 | 1 | 0.47 | 0.05 | 1 | 0.47 |
| DUALC1 | 0.03 | 1 | 0.04 | 0.03 | 1 | 0.05 | 0.01 | 2 | 0.03 | 0.01 | 2 | 0.02 | 0.01 | 1 | 0.02 |
| DUALC2 | 0.02 | 2 | 0.04 | 0.02 | 2 | 0.04 | 0.02 | 2 | 0.04 | 0.01 | 89 | 0.13 | 0.01 | 150 | 0.20 |
| DUALC5 | 0.03 | 2 | 0.06 | 0.03 | 2 | 0.06 | 0.02 | 2 | 0.05 | 0.01 | 9 | 0.04 | 0.01 | 26 | 0.07 |
| DUALC8 | 0.07 | 2 | 0.17 | 0.07 | 1 | 0.17 | 0.02 | 1 | 0.12 | 0.01 | 3 | 0.12 | 0.01 | 2 | 0.12 |
| GOULDQP2 | 0.48 | 2 | 0.72 | 0.47 | 2 | 0.78 | 0.26 | 4 | 0.45 | 0.12 | 1 | 0.20 | 0.12 | 1 | 0.21 |
| GOULDQP3 | 0.47 | 2 | 0.71 | 0.46 | 2 | 0.76 | 0.26 | 4 | 0.47 | 0.12 | 1 | 0.21 | 0.12 | 1 | 0.21 |
| KSIP | 0.50 | 2 | 1.03 | 0.50 | 1 | 1.03 | 0.03 | 1 | 0.56 | 0.01 | 2 | 0.54 | 0.01 | 1 | 0.52 |
| MOSARQP1 | 0.09 | 3 | 0.15 | 0.09 | 3 | 0.15 | 0.04 | 6 | 0.09 | 0.03 | 2 | 0.06 | 0.02 | 2 | 0.06 |
| NCVXQP1 | 0.41 | 2 | 0.69 | 0.46 | 2 | 0.79 | 0.14 | 2 | 0.32 | 0.06 | 1 | 0.21 | 0.06 | 1 | 0.21 |
| NCVXQP2 | 0.41 | 14 | 1.23 | 0.46 | 10 | 1.29 | 0.15 | 1 | 0.30 | 0.06 | 1 | 0.22 | 0.06 | 1 | 0.22 |
| NCVXQP3 | 0.41 | 42 | 2.65 | 0.46 | 15 | 1.59 | 0.13 | 2 | 0.32 | 0.06 | 1 | 0.24 | 0.06 | 1 | 0.22 |
| NCVXQP4 | 0.16 | 3 | 0.33 | 0.16 | 3 | 0.33 | 0.11 | 2 | 0.20 | 0.07 | 1 | 0.13 | 0.08 | 1 | 0.14 |
| NCVXQP5 | 0.17 | 13 | 0.66 | 0.17 | 13 | 0.66 | 0.15 | 4 | 0.27 | 0.09 | 1 | 0.15 | 0.07 | 1 | 0.13 |
| NCVXQP6 | 0.16 | 26 | 0.98 | 0.16 | 19 | 0.77 | 0.14 | 2 | 0.22 | 0.08 | 1 | 0.14 | 0.07 | 1 | 0.13 |
| NCVXQP7 | 0.83 | 2 | 1.40 | 527.57 | 3 | 529.12 | 0.19 | 1 | 0.63 | 0.05 | 1 | 0.46 | 0.07 | 1 | 0.47 |
| NCVXQP8 | 0.87 | 11 | 2.00 | 528.25 | 8 | 530.43 | 0.17 | 1 | 0.62 | 0.05 | 1 | 0.49 | 0.06 | 1 | 0.48 |
| NCVXQP9 | 0.72 | 66 | 4.99 | 539.13 | 16 | 542.62 | 0.16 | 1 | 0.58 | 0.05 | 1 | 0.47 | 0.06 | 1 | 0.48 |
| POWELL20 | 0.30 | 1 | 0.42 | 0.32 | 1 | 0.45 | 0.13 | 1 | 0.19 | 0.08 | 2 | 0.16 | 0.06 | 1 | 0.12 |
| PRIMAL1 | 0.12 | 2 | 0.13 | 0.12 | 1 | 0.13 | 0.04 | 18 | 0.07 | 0.02 | 9 | 0.04 | 0.02 | 2 | 0.03 |
| PRIMAL2 | 0.16 | 1 | 0.17 | 0.15 | 1 | 0.16 | 0.05 | 19 | 0.09 | 0.02 | 2 | 0.03 | 0.01 | 2 | 0.02 |
| PRIMAL3 | 0.58 | 1 | 0.60 | 0.58 | 1 | 0.61 | 0.03 | 24 | 0.13 | 0.02 | 2 | 0.04 | 0.02 | 2 | 0.05 |
| PRIMAL4 | 0.30 | 1 | 0.32 | 0.30 | 1 | 0.32 | 0.03 | 15 | 0.09 | 0.03 | 2 | 0.04 | 0.03 | 2 | 0.04 |
| PRIMALC1 | 0.02 | 1 | 0.02 | 0.02 | 1 | 0.02 | 0.02 | 3 | 0.03 | 0.02 | 248 | 0.25 | 0.01 | 97 | 0.11 |
| PRIMALC2 | 0.01 | 1 | 0.02 | 0.01 | 1 | 0.01 | 0.06 | 2 | 0.06 | 0.01 | 245 | 0.24 | 0.01 | 132 | 0.13 |
| PRIMALC5 | 0.02 | 1 | 0.02 | 0.03 | 1 | 0.03 | 0.03 | 3 | 0.03 | 0.02 | 7 | 0.03 | 0.01 | 8 | 0.02 |
| PRIMALC8 | 0.03 | 1 | 0.04 | 0.04 | 1 | 0.04 | 0.03 | 3 | 0.04 | 0.01 | 16 | 0.03 | 0.01 | 12 | 0.03 |
| QPBAND | 0.26 | 2 | 0.40 | 0.24 | 2 | 0.41 | 0.19 | 2 | 0.27 | 0.11 | 12 | 0.44 | 0.10 | 11 | 0.41 |
| QPNBAND | 0.24 | 1 | 0.33 | 0.24 | 2 | 0.42 | 0.19 | 2 | 0.27 | 0.10 | 10 | 0.39 | 0.12 | 8 | 0.34 |
| QPCBOEI1 | 0.05 | 6 | 0.10 | 0.05 | 4 | 0.10 | 0.03 | 2 | 0.06 | 0.01 | 2 | 0.04 | 0.01 | 2 | 0.04 |
| QPCBOEI2 | 0.10 | 6 | 0.12 | 0.10 | 3 | 0.12 | 0.02 | 2 | 0.03 | 0.01 | 1 | 0.02 | 0.01 | 1 | 0.02 |
| QPNBOEI1 | 0.05 | 7 | 0.11 | 0.05 | 5 | 0.11 | 0.04 | 2 | 0.07 | 0.02 | 2 | 0.05 | 0.01 | 2 | 0.05 |
| QPNBOEI2 | 0.11 | 6 | 0.13 | 0.09 | 3 | 0.11 | 0.02 | 2 | 0.03 | 0.01 | 1 | 0.02 | 0.01 | 1 | 0.02 |
| QPCSTAIR | 0.05 | 4 | 0.12 | 0.05 | 3 | 0.12 | 0.02 | 3 | 0.08 | 0.01 | 1 | 0.06 | 0.01 | 1 | 0.06 |
| QPNSTAIR | 0.06 | 4 | 0.12 | 0.06 | 3 | 0.12 | 0.02 | 4 | 0.09 | 0.01 | 1 | 0.06 | 0.01 | 1 | 0.06 |
| SOSQP1 | 0.11 | 1 | 0.16 | 0.12 | 1 | 0.18 | 0.07 | 1 | 0.10 | 0.03 | 1 | 0.06 | 0.03 | 1 | 0.06 |
| STCQP2 | 0.13 | 1 | 0.19 | 0.14 | 1 | 0.20 | 0.05 | 4 | 0.12 | 0.03 | 255 | 3.08 | 0.03 | 4 | 0.09 |
| STNQP2 | 0.31 | 1 | 0.43 | 0.32 | 5 | 0.58 | 0.12 | 4 | 0.28 | 0.06 | 1 | 0.14 | 0.06 | 5 | 0.23 |
| UBH1 | 0.34 | 1 | 0.52 | 0.34 | 1 | 0.52 | 0.14 | 2 | 0.29 | 0.05 | 1 | 0.17 | 0.05 | 4 | 0.23 |

Table C.4: CUTEr QP problems—residual decrease of at least $10^{-8}$ and $C$ given by (5.9)

| | Explicit factors | | | | | | Implicit factors | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $G = H$ | | | $G = I$ | | | Family 1 | | | Family 2(a) | | | Family 2(b) | | |
| name | fact. | iter. | total | fact. | iter. | total | fact. | iter. | total | fact. | iter. | total | fact. | iter. | total |
| AUG2DCQP | 0.13 | 6 | 0.26 | 0.13 | 7 | 0.27 | 0.06 | 163 | 1.12 | 0.02 | 1158 | 7.94 | 0.02 | 233 | 1.61 |
| AUG2DQP | 0.12 | 6 | 0.24 | 0.12 | 7 | 0.26 | 0.06 | 159 | 1.09 | 0.02 | 72 | 0.55 | 0.02 | 247 | 1.74 |
| AUG3DCQP | 0.16 | 7 | 0.32 | 0.17 | 7 | 0.33 | 0.06 | 109 | 0.79 | 0.03 | 88 | 0.69 | 0.03 | 65 | 0.53 |
| AUG3DQP | 0.17 | 8 | 0.34 | 0.17 | 8 | 0.34 | 0.06 | 107 | 0.75 | 0.03 | 91 | 0.71 | 0.03 | 65 | 0.52 |
| BLOCKQP1 | 4.94 | 2 | 33.05 | 4.93 | 2 | 32.97 | 0.14 | 1 | 28.12 | 0.06 | 2 | 28.18 | 0.07 | 2 | 28.17 |
| BLOCKQP2 | 4.93 | 2 | 33.02 | 4.87 | 2 | 32.90 | 0.14 | 1 | 28.07 | 0.07 | 2 | 28.17 | 0.07 | 2 | 28.14 |
| BLOCKQP3 | 4.94 | 2 | 32.92 | 4.81 | 3 | 32.82 | 0.14 | 1 | 28.11 | 0.06 | 2 | 28.09 | 0.06 | 2 | 28.16 |
| BLOWEYA | 0.25 | 5 | 0.36 | 14.64 | 4 | 14.82 | 0.06 | 1 | 0.09 | 0.02 | 1 | 0.06 | 0.02 | 1 | 0.06 |
| BLOWEYB | 0.26 | 5 | 0.37 | 14.72 | 4 | 14.90 | 0.06 | 1 | 0.09 | 0.03 | 1 | 0.06 | 0.02 | 1 | 0.06 |
| BLOWEYC | 0.26 | 5 | 0.37 | 12.99 | 4 | 13.18 | 0.06 | 1 | 0.10 | 0.03 | 1 | 0.07 | 0.03 | 1 | 0.06 |
| CONT-050 | 0.36 | 7 | 0.67 | 0.60 | 8 | 1.03 | 0.05 | 8 | 0.22 | 0.01 | 19 | 0.27 | 0.01 | 19 | 0.27 |
| CONT-101 | 2.17 | 6 | 3.96 | 4.33 | 7 | 6.97 | 0.20 | 0 | 1.02 | 0.05 | 0 | 0.86 | 0.05 | 0 | 0.84 |
| CONT-201 | 13.18 | 6 | 24.10 | 29.39 | 6 | 43.18 | 0.87 | 0 | 6.92 | 0.23 | 0 | 6.37 | 0.23 | 0 | 6.30 |
| CONT1-10 | 2.16 | 6 | 3.99 | 4.14 | 8 | 6.80 | 0.20 | 8 | 1.33 | 0.05 | 19 | 1.54 | 0.05 | 19 | 1.53 |
| CONT1-20 | 13.24 | 6 | 24.70 | 28.77 | 7 | 43.92 | 0.88 | 0 | 7.53 | 0.22 | 0 | 6.72 | 0.22 | 0 | 6.75 |
| CONT5-QP | ran out of memory | | | ran out of memory | | | 0.87 | 1 | 6.55 | 0.25 | 36 | 13.90 | 0.25 | 36 | 13.30 |
| CVXQP1 | 0.48 | 124 | 6.61 | 0.46 | 73 | 4.99 | 0.14 | 97 | 2.63 | 0.06 | 21 | 0.74 | 0.06 | 22 | 0.76 |
| CVXQP2 | 0.16 | 116 | 3.66 | 0.16 | 85 | 2.72 | 0.11 | 180 | 3.77 | 0.07 | 27 | 0.71 | 0.07 | 15 | 0.45 |
| CVXQP3 | 0.71 | 134 | 9.71 | 505.39 | 67 | 513.12 | 0.16 | 4 | 0.70 | 0.05 | 19 | 1.14 | 0.05 | 20 | 1.14 |
| DUALC1 | 0.03 | 8 | 0.05 | 0.03 | 7 | 0.05 | 0.01 | 11 | 0.05 | 0.01 | 130 | 0.18 | 0.01 | 13 | 0.04 |
| DUALC2 | 0.02 | 5 | 0.05 | 0.02 | 3 | 0.04 | 0.02 | 8 | 0.04 | 0.01 | 163 | 0.22 | 0.01 | 228 | 0.29 |
| DUALC5 | 0.03 | 10 | 0.07 | 0.03 | 8 | 0.07 | 0.02 | 9 | 0.06 | 0.01 | 13 | 0.05 | 0.01 | 55 | 0.11 |
| DUALC8 | 0.07 | 9 | 0.19 | 0.07 | 4 | 0.18 | 0.02 | 12 | 0.14 | 0.01 | 51 | 0.21 | 0.01 | 71 | 0.26 |
| GOULDQP2 | 0.48 | 5 | 0.95 | 0.47 | 5 | 1.01 | 0.26 | 18 | 0.91 | 0.12 | 42 | 1.68 | 0.12 | 30 | 1.21 |
| GOULDQP3 | 0.47 | 5 | 0.94 | 0.46 | 5 | 1.00 | 0.26 | 18 | 0.94 | 0.12 | 34 | 1.42 | 0.12 | 51 | 2.09 |
| KSIP | 0.50 | 6 | 1.05 | 0.50 | 8 | 1.07 | 0.03 | 15 | 0.61 | 0.01 | 6 | 0.57 | 0.01 | 5 | 0.54 |
| MOSARQP1 | 0.09 | 14 | 0.26 | 0.09 | 13 | 0.25 | 0.04 | 50 | 0.26 | 0.03 | 14 | 0.12 | 0.02 | 14 | 0.11 |
| NCVXQP1 | 0.41 | 9898 | 483.79 | 0.46 | 5925 | 352.31 | 0.14 | 91 | 2.46 | 0.06 | 21 | 0.73 | 0.06 | 22 | 0.74 |
| NCVXQP2 | 0.41 | 9929 | 465.07 | 0.46 | 9929 | 582.50 | 0.15 | 4966 | 120.38 | 0.06 | 23 | 0.78 | 0.06 | 23 | 0.78 |
| NCVXQP3 | 0.41 | 9997 | 466.65 | 0.46 | 8242 | 492.05 | 0.13 | 92 | 2.48 | 0.06 | 21 | 0.72 | 0.06 | 21 | 0.72 |
| NCVXQP4 | 0.16 | 9489 | 296.79 | 0.16 | 8756 | 277.30 | 0.11 | 2693 | 52.80 | 0.07 | 28 | 0.74 | 0.08 | 16 | 0.48 |
| NCVXQP5 | 0.17 | 9990 | 319.96 | 0.17 | 9973 | 320.34 | 0.15 | 9970 | 195.47 | 0.09 | 28 | 0.75 | 0.07 | 15 | 0.44 |
| NCVXQP6 | 0.16 | 7284 | 209.70 | 0.16 | 9835 | 287.43 | 0.14 | 4658 | 85.75 | 0.08 | 27 | 0.72 | 0.07 | 15 | 0.44 |
| NCVXQP7 | 0.83 | 9906 | 598.40 | 527.57 | 6192 | 1120.71 | 0.19 | 4 | 0.76 | 0.05 | 19 | 1.12 | 0.07 | 20 | 1.16 |
| NCVXQP8 | 0.87 | 9918 | 640.50 | 528.25 | 9756 | 1523.06 | 0.17 | 4 | 0.76 | 0.05 | 19 | 1.20 | 0.06 | 20 | 1.17 |
| NCVXQP9 | 0.72 | 9997 | 578.21 | 539.13 | 9884 | 1467.28 | 0.16 | 4 | 0.70 | 0.05 | 19 | 1.13 | 0.06 | 20 | 1.15 |
| POWELL20 | 0.30 | 1 | 0.41 | 0.32 | 1 | 0.46 | 0.13 | 1 | 0.19 | 0.08 | 317 | 5.01 | 0.06 | 10 | 0.26 |
| PRIMAL1 | 0.12 | 6 | 0.14 | 0.12 | 9 | 0.15 | 0.04 | 166 | 0.28 | 0.02 | 15 | 0.05 | 0.02 | 30 | 0.07 |
| PRIMAL2 | 0.16 | 6 | 0.19 | 0.15 | 7 | 0.18 | 0.05 | 133 | 0.31 | 0.02 | 23 | 0.08 | 0.01 | 11 | 0.04 |
| PRIMAL3 | 0.58 | 6 | 0.63 | 0.58 | 6 | 0.63 | 0.03 | 120 | 0.44 | 0.02 | 10 | 0.07 | 0.02 | 9 | 0.07 |
| PRIMAL4 | 0.30 | 5 | 0.34 | 0.30 | 5 | 0.34 | 0.03 | 62 | 0.25 | 0.03 | 8 | 0.07 | 0.03 | 7 | 0.06 |
| PRIMALC1 | 0.02 | 5 | 0.03 | 0.02 | 4 | 0.02 | 0.02 | 7 | 0.03 | 0.02 | 248 | 0.26 | 0.01 | 248 | 0.26 |
| PRIMALC2 | 0.01 | 4 | 0.02 | 0.01 | 4 | 0.02 | 0.06 | 6 | 0.06 | 0.01 | 245 | 0.25 | 0.01 | 12 | 0.02 |
| PRIMALC5 | 0.02 | 4 | 0.03 | 0.03 | 4 | 0.03 | 0.03 | 6 | 0.03 | 0.02 | 10 | 0.03 | 0.01 | 11 | 0.02 |
| PRIMALC8 | 0.03 | 5 | 0.05 | 0.04 | 4 | 0.05 | 0.03 | 6 | 0.04 | 0.01 | 21 | 0.04 | 0.01 | 56 | 0.10 |
| QPBAND | 0.26 | 5 | 0.54 | 0.24 | 4 | 0.51 | 0.19 | 8 | 0.40 | 0.11 | 489 | 12.58 | 0.10 | 141 | 3.66 |
| QPNBAND | 0.24 | 9 | 0.69 | 0.24 | 6 | 0.61 | 0.19 | 7 | 0.38 | 0.10 | 131 | 3.43 | 0.12 | 76 | 2.06 |
| QPCBOEI1 | 0.05 | 21 | 0.15 | 0.05 | 17 | 0.15 | 0.03 | 103 | 0.26 | 0.01 | 23 | 0.09 | 0.01 | 24 | 0.09 |
| QPCBOEI2 | 0.10 | 19 | 0.15 | 0.10 | 18 | 0.15 | 0.02 | 4 | 0.03 | 0.01 | 2 | 0.03 | 0.01 | 2 | 0.03 |
| QPNBOEI1 | 0.05 | 21 | 0.15 | 0.05 | 17 | 0.15 | 0.04 | 104 | 0.28 | 0.02 | 22 | 0.10 | 0.01 | 24 | 0.09 |
| QPNBOEI2 | 0.11 | 18 | 0.16 | 0.09 | 18 | 0.15 | 0.02 | 4 | 0.03 | 0.01 | 2 | 0.02 | 0.01 | 2 | 0.02 |
| QPCSTAIR | 0.05 | 20 | 0.17 | 0.05 | 19 | 0.18 | 0.02 | 137 | 0.34 | 0.01 | 15 | 0.09 | 0.01 | 970 | 2.03 |
| QPNSTAIR | 0.06 | 20 | 0.18 | 0.06 | 19 | 0.18 | 0.02 | 125 | 0.32 | 0.01 | 15 | 0.09 | 0.01 | 11 | 0.09 |
| SOSQP1 | 0.11 | 2 | 0.18 | 0.12 | 2 | 0.20 | 0.07 | 5 | 0.15 | 0.03 | 8 | 0.15 | 0.03 | 59 | 0.77 |
| STCQP2 | 0.13 | 57 | 1.07 | 0.14 | 51 | 1.14 | 0.05 | 67 | 0.79 | 0.03 | 6029 | 71.04 | 0.03 | 5989 | 66.54 |
| STNQP2 | 0.31 | 2402 | 87.38 | 0.32 | 529 | 18.89 | 0.12 | 930 | 22.62 | 0.06 | 1 | 0.14 | 0.06 | 5 | 0.24 |
| UBH1 | 0.34 | 6 | 0.76 | 0.34 | 5 | 0.67 | 0.14 | 28 | 0.82 | 0.05 | 31 | 0.81 | 0.05 | 24 | 0.65 |