Science & Technology Facilities Council

Technical Report
RAL-TR-2013-003

Iterative methods for symmetric quasi-definite linear systems Part I: Theory

M Arioli, D Orban

May 2013

RAL Library
STFC Rutherford Appleton Laboratory
R61
Harwell Oxford
Didcot
OX11 0QX

Tel: +44(0)1235 445384
Fax: +44(0)1235 446403
email: libraryral@stfc.ac.uk

Science and Technology Facilities Council reports are available online at: http://epubs.stfc.ac.uk

ISSN 1358- 6254

ITERATIVE METHODS FOR SYMMETRIC QUASI-DEFINITE LINEAR SYSTEMS PART I: THEORY

MARIO ARIOLI AND DOMINIQUE ORBAN

Abstract. Symmetric quasi-definite systems may be interpreted as regularized linear least-squares problem in appropriate metrics and arise from applications such as regularized interior-point methods for convex optimization and stabilized control problems. We propose two families of Krylov methods well suited to the solution of such systems based on a preconditioned variant of the Golub-Kahan bidiagonalization process. The first family contains methods operating of the normal and Schur-complement equations, including generalizations of well-known methods such as Lsqr and Lsmr but also a new method named Craig-mr aiming to minimize the residual of the Schur-complement equations. The second family follows from a related Lanczos process and contains methods operating directly on the augmented system, which generalize the conjugate-gradient and minimum-residual methods. We establish connections between augmented-system and reduced-system methods. In particular, the conjugate-gradient method is well defined despite the indefiniteness of the operator. We provide an explanation for the often-observed staircase behavior of the residual in the minimum-residual method. An additional contribution is to provide explicit stopping criteria for all methods based on estimates of the relative direct error in appropriate norms, as opposed to criteria based on the residual. A lower bound estimate is available at no additional computational cost while an upper bound estimate comes at the cost of a few additional scalar operations per iteration.

Date: May 16, 2013.

2010 *Mathematics Subject Classification.* 90C06, 90C20, 90C30, 90C51, 90C53, 90C55, 65F10, 65F50.

Key words and phrases. Symmetric quasi-definite system, generalized Golub-Kahan bidiagonalization, elliptic singular values, linear least-squares problem, Lsqr, Craig, Lsmr, Craig-mr, Cg, Minres, Lanczos.

Research supported by EPSRC Grant EP/E053351/1.

Research partially supported by NSERC Discovery Grant 299010-04.

Contents

List of Algorithms

List of Figures

1. Introduction

Symmetric quasi-definite (SQD) linear systems have the general form

$$\begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & -\mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}, \tag{1.1}$$

where $\mathbf{M} \in \mathbb{R}^{n\times n}$ and $\mathbf{N} \in \mathbb{R}^{m\times m}$ are symmetric and positive definite. The coefficient matrix of (1.1) is then itself said to be SQD. It is always symmetric and indefinite unless $m = 0$ or $n = 0$, in which case it is definite. Vanderbei (1995) shows that SQD matrices are *strongly factorizable*, i.e., any symmetric permutation of their rows and columns admits a Cholesky-like factorization without pivoting. The latter factorization can therefore typically be computed in much less operations than a traditional symmetric indefinite factorization such as that of Duff (2004) and often has sparser factors. We adopt the following definition of an SQD matrix.

Definition 1.1. A matrix $\mathbf{K} \in \mathbb{R}^{(n+m)\times(n+m)}$ is said to be symmetric quasi-definite (SQD) if $\mathbf{K} = \mathbf{K}^T$ and there exists a permutation matrix $\mathbf{P} \in \mathbb{R}^{(n+m)\times(n+m)}$ such that $\mathbf{P}^T\mathbf{K}\mathbf{P}$ has the form (1.1).

Among other important properties of (1.1) are that the system is always square, symmetric, indefinite and nonsingular, irrespective of the rank of $\mathbf{A}$, and the inverse of the coefficient matrix is itself SQD. For more details, we refer to (Vanderbei, 1995).

In this paper, we devise iterative methods for the solution of (1.1) that exploit its structure. Our methods are generalizations of Lsqr (Paige and Saunders, 1982), Craig (Craig, 1955) and Lsmr (Fong and Saunders, 2011) based on a Golub-Kahan process (Golub and Kahan, 1965) occurring in the appropriate metric. In addition, we present a method named Craig-mr aiming to minimize the residual of the Schur-complement equations. Those methods determine specialized implementations of Cg (Hestenes and Stiefel, 1952) and Minres (Paige and Saunders, 1975) by identifying specialized Lanczos processes. The implementation of Cg is particularly interesting given the indefiniteness of (1.1).

All methods presented here especially apply to cases where systems with coefficient matrices $\mathbf{M}$ and $\mathbf{N}$ can be solved efficiently. In follow-up research, we investigate preconditioning strategies and error analyses related to cases where this assumption is not satisfied. All methods methods discussed below are dimension agnostic in the sense that each applies irrespective of the fact that $m < n$ or $m \geq n$.

Systems of the form (1.1) arise in numerous applications and their efficient iterative solution is crucial to matrix-free methods. In interior-point methods for the optimization of the inequality-constrained convex problem

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \;\; f(\mathbf{x}) \quad \text{subject to} \;\; c(\mathbf{x}) \geq 0,$$

where f is convex and c is concave, Newton-based methods typically solve direction-finding systems of the form

$$\begin{bmatrix} \nabla_{\mathbf{xx}} L(\mathbf{x},\mathbf{y}) & J(\mathbf{x})^T \\ J(\mathbf{x}) & -Y^{-1}C(\mathbf{x}) \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ -\Delta\mathbf{y} \end{bmatrix} = -\begin{bmatrix} \nabla_{\mathbf{x}} L(\mathbf{x},\mathbf{y}) \\ c(\mathbf{x}) - \mu Y^{-1}\mathbf{e} \end{bmatrix},$$

where $L(\mathbf{x},\mathbf{y}) = f(\mathbf{x}) - c(\mathbf{x})^T\mathbf{y}$ is the Lagrangian of the problem, $J(\mathbf{x})$ is the Jacobian of the constraints, $Y = \operatorname{diag}(\mathbf{y})$, $C(\mathbf{x}) = \operatorname{diag}(c(\mathbf{x}))$, $\mu > 0$ is a parameter, and $(c(\mathbf{x}),\mathbf{y}) > 0$ is enforced. Whenever either f is strictly convex or $J(\mathbf{x})$ has

full row rank, the leading block is positive definite and the system above is SQD. When either of those assumptions is not satisfied, regularized methods such as that of Friedlander and Orban (2012) recovers an SQD system, even in the presence of linear equality constraints.

Regularized linear least-squares problems are often cast as sytems of the form (1.1) in which $\mathbf{M}$ and $\mathbf{N}$ are multiples of the identity. The right-hand side in this case is typically of the form $(\mathbf{b}, \mathbf{0})$ or $(\mathbf{0}, \mathbf{b})$. With such a right-hand side, the methods we propose below are in fact an interpretation of (1.1) as a regularized linear least-squares problem in a non-Euclidian metric.

The mixed finite-element approximation to the globally stabilized Stokes problem in weak form can be stated as

$$\begin{aligned} \int_\Omega \nabla u : \nabla v \,\mathrm{d}x - \int_\Omega p \,\nabla \cdot v \,\mathrm{d}x &= \int_\Omega f v \,\mathrm{d}x, \quad \text{for all } v \in V, \\ -\int_\Omega q \,\nabla \cdot u \,\mathrm{d}x - \beta \mathcal{C}(p,q) &= 0 \qquad\qquad \text{for all } q \in Q, \end{aligned} \tag{1.2}$$

where u is the velocity field, p is the pressure field, Ω is the domain, ":" represents the componentwise inner product, V and Q are compatible finite-dimensional function subspaces of test functions, $\beta > 0$ is a stabilization parameter and $\mathcal{C}$ is a stabilization term. After discretization with, e.g., P_1–P_1 triangular elements and continuous linear pressure, the above equations reduce to a linear system of the form (1.1) where $\mathbf{g} = \mathbf{0}$, $\mathbf{A}$ is the gradient matrix, $\mathbf{A}^T$ is the divergence matrix, $\mathbf{M}$ is the vector-Laplacian matrix and $\mathbf{N}$ represents the stabilization term. For more details, we refer the interested reader to (Silvester and Wathen, 1994).

Other applications of regularized linear least squares include Kalman (1960) filters (Bunse-Gertner, 2012; Strang, 1986) and variational data assimilation (Courtier, 1997). The problem formulation in both applications is very similar. The incremental formulation of the three-dimensional variational data assimilation problem may be stated as

$$\underset{\Delta\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \;\; \tfrac{1}{2}\|\Delta\mathbf{x}\|^2_{\mathbf{B}^{-1}} + \tfrac{1}{2}\|\mathbf{H}\Delta\mathbf{x} - \mathbf{d}\|^2_{\mathbf{R}^{-1}},$$

where $\Delta\mathbf{x} = \mathbf{x}_0 - \mathbf{x}_b$ is referred to as an *increment* used to obtain an initial climatic model state $\mathbf{x}_0$ from a *background* state x_b—i.e., a state resulting of previous forecasts, $\mathbf{B}$ is the covariance matrix of background error, $\mathbf{R}$ is the covariance matrix of observation errors, $\mathbf{H}$ represents a linearization of the observation operator and $\mathbf{d} = \mathbf{y}_0 - \mathbf{H}\mathbf{x}_b$ is the *innovation* vector, in which $\mathbf{y}_0$ is the observation vector. The optimality conditions of this problem have precisely the form (1.1) with $\mathbf{A} = \mathbf{H}$, $\mathbf{M} = \mathbf{R}$, $\mathbf{N} = \mathbf{B}^{-1}$, $\mathbf{f} = \mathbf{d}$ and $\mathbf{g} = \mathbf{0}$.

SQD systems have been used in the past to precondition standard symmetric saddle-point systems, i.e., for which $\mathbf{N} = 0$ (Axelson and Neytcheva, 2003; Perugia and Simoncini, 2000). Benzi, Golub, and Liesen (2005, Section 10.2, pp. 82–83) provide several references and summarize key results including eigenvalue estimates of the preconditioned system.

Notation. Throughout the paper, vectors and matrices are typeset in boldface while scalars appear in lightface. We use the notation $\mathbf{I}_k$ to denote the k-by-k identity matrix. For conciseness and when the context leaves no possible ambiguity, we will simply denote by $\mathbf{I}$ the identity matrix of appropriate size. For a symmetric positive definite $n \times n$ matrix $\mathbf{C}$, let $\|\mathbf{u}\|^2_{\mathbf{C}} = \mathbf{u}^T\mathbf{C}\mathbf{u} = \|\mathbf{C}^{\frac{1}{2}}\mathbf{u}\|^2_2$ be the norm defined

by $\mathbf{C}$, where $\mathbf{C}^{\frac{1}{2}}$ is the unique symmetric positive definite matrix square root of $\mathbf{C}$. Finally, the shorthand $\operatorname{blkdiag}(\mathbf{C}, \mathbf{D})$ is used to denote the block-diagonal matrix

$$\begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix}$$

for any matrices $\mathbf{C}$ and $\mathbf{D}$ of appropriate size. For any $x \in \mathbb{R}$, $\lfloor x \rfloor$ denotes the largest integer $k \leq x$ and $\lceil x \rceil$ denotes the smallest integer $k \geq x$.

Related Work. Existing iterative methods for symmetric indefinite systems, such as MINRES, SYMMLQ (Paige and Saunders, 1975) and SYMMBK (Chandra, 1978) do not exploit the rich quasi-definite structure of (1.1).

Assume we are able to factor $\mathbf{N} = \mathbf{L}\mathbf{D}\mathbf{L}^T$. Upon introducing auxiliary variables $\mathbf{z} := -\mathbf{D}\mathbf{L}^T\mathbf{y}$, it is possible to reformulate (1.1) as the traditional saddle-point point system

$$\begin{bmatrix} \mathbf{M} & & \mathbf{A} \\ & \mathbf{D}^{-1} & \mathbf{L}^T \\ \mathbf{A}^T & \mathbf{L} & \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \\ \mathbf{g} \end{bmatrix}.$$

Dollar et al. (2006) propose to solve the resulting system by way of the projected conjugate gradient method. This approach requires a projection into the nullspace of $\begin{bmatrix} \mathbf{A}^T & \mathbf{L} \end{bmatrix}$ at each iteration, which may be achieved via a one-time factorization of a projection matrix of the form

$$\begin{bmatrix} \tilde{\mathbf{M}} & & \mathbf{A} \\ & \tilde{\mathbf{D}}^{-1} & \mathbf{L}^T \\ \mathbf{A}^T & \mathbf{L} & \end{bmatrix},$$

for appropriate approximations $\tilde{\mathbf{M}} \approx \mathbf{M}$ and $\tilde{\mathbf{D}} \approx \mathbf{D}$ such that the above matrix has precisely m negative and $n + m$ positive eigenvalues.

Saunders (1995) derives extended versions of LSQR and CRAIG for the case where $\mathbf{N} = \lambda \mathbf{I}_m$ for some $\lambda \neq 0$ and establishes a connection with CG applied to the corresponding SQD system.

Gill, Saunders, and Shinnerl (1996) provide stability results for the LDL^T factorization of SQD matrices. George et al. (2000) and George and Ikramov (2000) examine additional properties of SQD matrices, of their eigenvalues and their condition number. Korzak (1999) gives the precise spectrum in the case of matrices arising from linear programming.

Benbow (1999) proposes a variant of LSQR similar to what we propose in the sequel of the present paper, only for the case where $\mathbf{N} = 0$ and systems with $\mathbf{M}$ can be solved efficiently.

The definite reference on solution methods for saddle-point linear systems is given by Benzi, Golub, and Liesen (2005). Although they mention SQD systems, no specialized iterative approach is suggested.

Marcia (2008) proposes a Kylov-type iterative method for general symmetric indefinite systems based on a symmetric indefinite factorization of the tridiagonal matrix generated by a Lanczos process. His method reduces to CG when applied to definite systems and is provably stable on indefinite systems.

Arioli (2010) derives a version of CRAIG for indefinite systems where $\mathbf{N} = 0$ based on the so-called *elliptic singular values* of $\mathbf{A}$.

The rest of this paper is organized as follows. Section 2 points out connections between various linear least-squares problems and (1.1) and §3 sets the prerequisites for the remainder of the paper. Section 4 defines the generalized Golub-Kahan process that forms the basis of our iterative methods. This process gives rise to generalized versions of LSQR in §6.1, CRAIG in §6.3 and LSMR in §6.5, as well as to a new method named G-CRAIG-MR in §6.7. Section 8.1 defines Lanczos processes determined by the generalized Golub-Kahan process and used to connect the previous methods to the method of conjugate gradients in §8.2 and to MINRES in §8.4. In particular, we demonstrate that the conjugate gradient method is well defined for SQD systems and solves a min-max problems. We also provide an explanation for the often-observed staircase behavior of the MINRES residual on symmetric saddle-point systems together with a description of what MINRES minimizes during iterations where the residual decreases and during iterations where the residual appears to plateau. We discuss our implementation in §9 and present numerical results on problems arising from optimization and discretized PDEs. We conclude with a discussion in §10.

2. Linear Least-Squares Problems

In this section, we recall the connections between various symmetric indefinite and SQD linear systems, and the solution of certain linear least-squares problems. Note that most of the connections are known. They are repeated here because they are the motivation for the derivation of iterative methods for (1.1). By convexity, all optimality conditions mentioned below are both necessary and sufficient.

The optimality conditions of the $\mathbf{M}^{-1}$-norm least-squares problem[1]

$$\underset{\mathbf{y}\in\mathbb{R}^m}{\text{minimize}} \ \tfrac{1}{2}\|\mathbf{A}\mathbf{y}-\mathbf{b}\|^2_{\mathbf{M}^{-1}}, \tag{2.1}$$

may be written

$$\begin{bmatrix}\mathbf{M} & \mathbf{A}\\ \mathbf{A}^T & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} = \begin{bmatrix}\mathbf{b}\\ \mathbf{0}\end{bmatrix}, \tag{2.2}$$

where $\mathbf{x} = \mathbf{M}^{-1}(\mathbf{b}-\mathbf{A}\mathbf{y})$ is the residual. The coefficient matrix of (2.2) is symmetric indefinite, but not SQD. Upon writing the Cholesky decomposition $\mathbf{M} = \mathbf{L}\mathbf{L}^T$, (2.2) are also the optimality conditions of both of the following *weighted*, or *preconditioned* least-squares problems

$$\underset{\mathbf{y}\in\mathbb{R}^m}{\text{minimize}} \ \tfrac{1}{2}\|\mathbf{L}^{-1}(\mathbf{A}\mathbf{y}-\mathbf{b})\|_2^2, \qquad \text{and} \qquad \underset{\mathbf{y}\in\mathbb{R}^m}{\text{minimize}} \ \tfrac{1}{2}\|\mathbf{M}^{-\frac{1}{2}}(\mathbf{A}\mathbf{y}-\mathbf{b})\|_2^2.$$

Whenever $\mathbf{A}$ does not have full column rank, (2.2) is singular. A typical remedy is to *regularize* the least-square problem, i.e., to change (2.1) to

$$\underset{\mathbf{y}\in\mathbb{R}^m}{\text{minimize}} \ \tfrac{1}{2}\left\|\begin{bmatrix}\mathbf{A}\\ \mathbf{R}\end{bmatrix}\mathbf{y} - \begin{bmatrix}\mathbf{b}\\ \mathbf{0}\end{bmatrix}\right\|^2_{\mathbf{M}_+^{-1}}, \qquad \text{where} \qquad \mathbf{M}_+ := \begin{bmatrix}\mathbf{M} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}_m\end{bmatrix} \tag{2.3}$$

and $\mathbf{R}$ is a square nonsingular matrix of appropriate size. Typically, $\mathbf{R} = \lambda\mathbf{I}_m$, for some regularization parameter $\lambda \in \mathbb{R}$ but other choices are possible. Note that the objective of (2.3) may equivalently be written

$$\tfrac{1}{2}\|\mathbf{A}\mathbf{y}-\mathbf{b}\|^2_{\mathbf{M}^{-1}} + \tfrac{1}{2}\|\mathbf{R}\mathbf{y}\|_2^2 = \tfrac{1}{2}\|\mathbf{A}\mathbf{y}-\mathbf{b}\|^2_{\mathbf{M}^{-1}} + \tfrac{1}{2}\|\mathbf{y}\|^2_{\mathbf{R}^T\mathbf{R}}.$$

[1] Sometimes referred to as the *generalized* least-squares problem.

The optimality conditions of (2.3) may then be written as the general SQD system

$$\begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & -\mathbf{R}^T\mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}.$$

Similarly, a positive-definite matrix of the form $\mathbf{N}^{\frac{1}{2}}$ can be used in place of $\mathbf{R}$ and this leads to (1.1) with $\mathbf{f} = \mathbf{b}$ and $\mathbf{g} = \mathbf{0}$. Additionally, (1.1) with $\mathbf{f} = \mathbf{b}$ and $\mathbf{g} = \mathbf{0}$ represents the optimality conditions of the $\mathbf{E}_+^{-1}$-norm regularized problem

$$\underset{\mathbf{y}\in\mathbb{R}^m}{\text{minimize}} \ \tfrac{1}{2} \left\| \begin{bmatrix} \mathbf{A} \\ \mathbf{I}_m \end{bmatrix} \mathbf{y} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_{\mathbf{E}_+^{-1}}^2, \qquad \text{where} \qquad \mathbf{E}_+ := \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{N}^{-1} \end{bmatrix}. \tag{2.4}$$

Equivalently, if $\mathbf{M} = \mathbf{L}\mathbf{L}^T$ and $\mathbf{N} = \mathbf{R}^T\mathbf{R}$, (1.1) represent the optimality conditions of the weighted regularized problem

$$\underset{\mathbf{y}\in\mathbb{R}^m}{\text{minimize}} \ \tfrac{1}{2} \left\| \begin{bmatrix} \mathbf{L}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \left(\begin{bmatrix} \mathbf{A} \\ \mathbf{I}_m \end{bmatrix} \mathbf{y} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right) \right\|_2^2. \tag{2.5}$$

A similar interpretation is derived when $\mathbf{L}$ and $\mathbf{R}$ are replaced by $\mathbf{M}^{\frac{1}{2}}$ and $\mathbf{N}^{\frac{1}{2}}$, respectively.

At this point it might seem attractive to simply employ LSQR to solve either (2.4) or (2.5). This would however require that we either compute $\mathbf{L}$ or solve systems with $\mathbf{M}^{\frac{1}{2}}$, and that we compute one of $\mathbf{R}$ and $\mathbf{N}^{\frac{1}{2}}$. Moreover, those must be computed accurately. Fortunately, there is an alternative in applications where solving systems with coefficient matrices $\mathbf{M}$ and $\mathbf{N}$ can be performed efficiently. We now give a few examples of such situations that arise in practice.

In regularized interior-point methods for linear programming—see, e.g., (Friedlander and Orban, 2012)—the matrices $\mathbf{M}$ and $\mathbf{N}$ are diagonal and solving systems with those matrices is therefore trivial. In interior-point methods for nonlinear programming in which the Hessian of the Lagrangian is approximated by a limited-memory quasi-Newton matrix in inverse form, solving systems with $\mathbf{M}$ is cheap since the limited-memory approximation represents $\mathbf{M}^{-1}$. A matrix-vector product is thus all that is required. Similarly, if the limited-memory quasi-Newton matrix is updated in factored form, solving systems with $\mathbf{M}$ is cost effective.

In fluid dynamics applications, the discretization of Darcy's law for incompressible flow in a saturated medium gives rise to a badly scaled matrix $\mathbf{M}$ but for which diagonal preconditioning will only leave a few clusters of eigenvalues independently of the mesh size (Wathen, 1987). It can thus be expected that the conjugate gradient method with diagonal preconditioner will converge quickly.

Without loss of generality, we assume from now on that the right-hand side of (1.1) has $\mathbf{f} = \mathbf{b}$ and $\mathbf{g} = \mathbf{0}$. Reduction to this situation is always possible, though admittedly at some cost, by first finding any $(\mathbf{x}^0, \mathbf{y}^0)$ satisfying $\mathbf{A}\mathbf{x}^0 - \mathbf{N}\mathbf{y}^0 = \mathbf{g}$ and setting $\mathbf{b} = \mathbf{f} - \mathbf{M}\mathbf{x}^0 - \mathbf{A}^T\mathbf{y}^0$. A variety of iterative methods can be used to identify such $(\mathbf{x}^0, \mathbf{y}^0)$. For instance, the minimum norm problem

$$\underset{\mathbf{x},\mathbf{y}}{\text{minimize}} \ \tfrac{1}{2} \left(\|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_2^2 \right) \quad \text{subject to} \ \mathbf{A}\mathbf{x} - \mathbf{N}\mathbf{y} = \mathbf{g}$$

can be solved with either the standard LSQR (Paige and Saunders, 1982) or CRAIG (Craig, 1955). Because of our assumption that solving systems with $\mathbf{N}$ can be done easily and efficiently, a simpler solution consists in setting $\mathbf{x}^0 := \mathbf{0}$ and solving $\mathbf{N}\mathbf{y}^0 = -\mathbf{g}$ for $\mathbf{y}^0$.

Consider (1.1) with $\mathbf{f} = \mathbf{b}$ and $\mathbf{g} = \mathbf{0}$. Upon eliminating $\mathbf{x}$ from the first equation, the $\mathbf{y}$ component of the solution must satisfy

$$(\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A} + \mathbf{N})\mathbf{y} = \mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{b}. \tag{2.6}$$

We refer to (2.6) as the *normal equations*. Similarly, eliminating $\mathbf{y}$ from the second equation of (1.1), the $\mathbf{x}$ component must satisfy

$$(\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}} + \mathbf{M})\mathbf{x} = \mathbf{b}, \tag{2.7}$$

to which we refer as the *Schur-complement equations*. Those equations are not, strictly speaking, normal equations as they do not directly describe the optimality conditions of a linear least-squares problem. As becomes apparent in later sections, the coefficient matrices of (2.6) and (2.7) play an important role in our iterative methods in that they define energy norms suitable to measure direct errors.

We close this section by mentioning that (2.4) is always equivalent to the least-norm problem

$$\underset{\mathbf{x},\mathbf{y}}{\text{minimize}} \ \tfrac{1}{2}(\|\mathbf{x}\|_{\mathbf{M}}^2 + \|\mathbf{y}\|_{\mathbf{N}}^2) \quad \text{subject to } \mathbf{M}\mathbf{x} + \mathbf{A}\mathbf{y} = \mathbf{b}. \tag{2.8}$$

Indeed the Lagrange multipliers associated to the equality constraints are precisely equal to $\mathbf{x}$. It is only when regularization is present that the least-squares problem and the least-norm problems are equivalent.

3. Preliminaries

3.1. **The Lanczos Process.** In the sequel, we often refer to Lanczos processes in various contexts. To establish the notation, consider a generic symmetric linear system with coefficient matrix $\mathbf{H}$ and right-hand side $\mathbf{d}$. A Lanczos (1950, 1952) process applied to $\mathbf{H}$ and $\mathbf{d}$ constructs a sequence of vectors $\{\mathbf{s}_k\}$ according to the following recursion:

$$\begin{aligned} \omega_1\mathbf{s}_1 &= \mathbf{d}, \\ \omega_{k+1}\mathbf{s}_{k+1} &= \mathbf{H}\mathbf{s}_k - \chi_k\mathbf{s}_k - \omega_k\mathbf{s}_{k-1}, \qquad \chi_k := \mathbf{s}_k^{\mathsf{T}}\mathbf{H}\mathbf{s}_k, \end{aligned} \tag{3.1}$$

with the convention $\mathbf{s}_0 := \mathbf{0}$. The constants ω_k are chosen at each iteration so $\|\mathbf{s}_k\|_2 = 1$. The notation used in (3.1) is intentionally non-standard so as to avoid confusion with scalar and vector quantities defined in the rest of the paper, the latter adhering more closely to standard notation found in the literature. It is possible to derive (3.1) from the standard Gram-Schmidt orthogonalization process, by exploiting symmetry of $\mathbf{H}$, and therefore, the vectors $\mathbf{s}_k$ are theoretically orthonormal. After k iterations, the Lanczos process has generated $k+1$ vectors. We gather the first k Lanczos vectors into the matrix $\mathbf{S}_k := \begin{bmatrix}\mathbf{s}_1 & \cdots & \mathbf{s}_k\end{bmatrix}$. The situation after k iterations may be summarized as

$$\mathbf{H}\mathbf{S}_k = \mathbf{S}_k\mathbf{\Omega}_k + \omega_{k+1}\mathbf{s}_{k+1}\mathbf{e}_k^{\mathsf{T}}, \tag{3.2}$$

where $\mathbf{\Omega}_k$ is the tridiagonal matrix

$$\mathbf{\Omega}_k := \begin{bmatrix} \chi_1 & \omega_2 & & \\ \omega_2 & \chi_2 & \ddots & \\ & \ddots & \ddots & \omega_k \\ & & \omega_k & \chi_k \end{bmatrix}.$$

In floating-point arithmetic, orthogonality of the vectors $\mathbf{s}_k$ is soon compromised and it is only mathematically that we are allowed to expect that

$$\mathbf{S}_k^{\mathsf{T}}\mathbf{H}\mathbf{S}_k = \mathbf{\Omega}_k,$$

but (3.2) generally holds to within machine precision.

3.2. **Hilbert Space Setting.** Let $\mathbf{H} \in \mathbb{R}^{k\times k}$ be symmetric and positive definite. Then $\mathbb{R}^k$ endowed with the scalar product $\mathbf{u}^{\mathsf{T}}\mathbf{H}\mathbf{v}$ is a Hilbert space. Conversely, let $\mathbb{H}$ be a k-dimensional Hilbert space with basis $\{\phi_j\}_{j=1,\dots,k}$ and equipped with a scalar product $(u,v)_{\mathbb{H}}$. Then $\mathbb{H}$ is isometric to $\mathbb{R}^k$ with a scalar product determined by the Gramian matrix $\mathbf{H}$, i.e., $\mathbf{H}_{ij} := (\phi_i,\phi_j)_{\mathbb{H}}$. Indeed, upon decomposing $u = \sum_j u_j\phi_j$ and $v = \sum_j v_j\phi_j$, we have $(u,v)_{\mathbb{H}} = \mathbf{u}^{\mathsf{T}}\mathbf{H}\mathbf{v}$. Owing to the Riesz theorem (Brézis, 1983), the dual space $\mathbb{H}^\star$ of $\mathbb{H}$ is itself a Hilbert space with a scalar product induced by $\mathbf{H}^{-1}$. In particular, the operator

$$\mathscr{H} : \mathbb{H} \to \mathbb{H}^\star \qquad \langle \mathscr{H}u, v\rangle_{\mathbb{H}^\star,\mathbb{H}} := \mathbf{v}^{\mathsf{T}}\mathbf{H}\mathbf{u}$$

is self-adjoint and strictly positive, and therefore invertible. Furthermore, the basis $\{\phi_i\}$ is made of the columns of $\mathbf{H}$ and the corresponding basis $\{\psi_i\}$ of $\mathbb{H}^\star$ is made of the columns of $\mathbf{H}^{-1}$. Hereafter, all our Hilbert spaces are finite dimensional.

Given $z \in \mathbb{H}^\star$, we have

$$\langle z,u\rangle_{\mathbb{H}^\star,\mathbb{H}} = \mathbf{z}^{\mathsf{T}}\mathbf{u} = \mathbf{z}^{\mathsf{T}}\mathbf{H}^{-1}\mathbf{H}\mathbf{u} = (\mathbf{u},\mathbf{H}^{-1}\mathbf{z})_{\mathbb{H}},$$

and we have that $\mathbf{w} = \mathbf{H}^{-1}\mathbf{z}$ is the representation of the Riesz vector $w = \sum_j w_j\phi_j \in \mathbb{H}$. Let $\mathscr{K} : \mathbb{H} \to \mathbb{F}$ be an operator between the Hilbert spaces $\mathbb{H}$ and $\mathbb{F}$. Its adjoint operator $\mathscr{K}^\star : \mathbb{F}^\star \to \mathbb{H}^\star$ is defined (Brézis, 1983) by

$$\langle \mathscr{K}^\star v, u\rangle_{\mathbb{H}^\star,\mathbb{H}} := \langle v, \mathscr{K}u\rangle_{\mathbb{F}^\star,\mathbb{F}} \quad \forall v \in \mathbb{F}^\star, u \in \mathbb{H}.$$

Therefore, we have

$$\langle \mathscr{K}^\star v, u\rangle_{\mathbb{H}^\star,\mathbb{H}} = (\mathbf{H}^{-1}\mathbf{v}, \mathbf{K}\mathbf{u})_{\mathbb{H}} = \mathbf{u}^{\mathsf{T}}\mathbf{K}^{\mathsf{T}}\mathbf{v}, \tag{3.3}$$

where $\mathbf{K}$ is a matrix representation of $\mathscr{K}$. Finally, if we assume that $\mathbb{F} = \mathbb{H}^\star$ then the "normal equations operator" is

$$\mathscr{K}^\star \circ \mathscr{H}^{-1} \circ \mathscr{K} : \mathbb{H} \to \mathbb{H}^\star,$$

and it is represented by the matrix $\mathbf{K}^{\mathsf{T}}\mathbf{H}^{-1}\mathbf{K}$. If $\mathbf{K}^{\mathsf{T}} = \mathbf{K}$ then $\mathscr{K}$ is self-adjoint. Moreover, the operator

$$\mathscr{H}^{-1} \circ \mathscr{K} : \mathbb{H} \to \mathbb{H} \tag{3.4}$$

maps $\mathbb{H}$ into itself. Therefore, we can define its powers $(\mathscr{H}^{-1}\circ\mathscr{K})^i$ as the operators represented by the matrices $(\mathbf{H}^{-1}\mathbf{K})^i$ for $i \geq 0$.

Let us consider now the Hilbert spaces

$$\mathbb{M} := (\mathbb{R}^n, \|\cdot\|_{\mathbf{M}}), \qquad \mathbb{N} := (\mathbb{R}^m, \|\cdot\|_{\mathbf{N}}),$$

their duals

$$\mathbb{M}^\star := (\mathbb{R}^n, \|\cdot\|_{\mathbf{M}^{-1}}), \qquad \mathbb{N}^\star := (\mathbb{R}^m, \|\cdot\|_{\mathbf{N}^{-1}}),$$

and assume that

$$\mathbf{K} := \begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^{\mathsf{T}} & -\mathbf{N} \end{bmatrix} \tag{3.5}$$

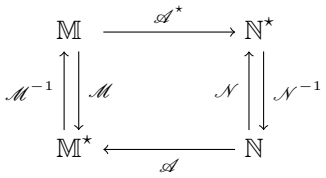

FIGURE 3.1. Commutative diagram between the relevant Hilbert spaces.

with the corresponding operator

$$\mathscr{K} : \mathbb{M} \times \mathbb{N} \to \mathbb{M}^\star \times \mathbb{N}^\star.$$

The norm and scalar product in $\mathbb{M} \times \mathbb{N}$ are induced by the block-diagonal matrix

$$\mathbf{H} = \begin{bmatrix} \mathbf{M} & \\ & \mathbf{N} \end{bmatrix}. \tag{3.6}$$

Let the matrix $\mathbf{A}$ represent the linear operator $\mathscr{A} : \mathbb{N} \to \mathbb{M}^\star$ with respect to the canonical bases. For $y \in \mathbb{N}$, $\mathscr{A}y$ may be considered as a linear operator defined on $\mathbb{M}$ via the relation

$$\langle \mathscr{A}y, u \rangle_{\mathbb{M}^\star,\mathbb{M}} := (\mathbf{u}, \mathbf{M}^{-1}\mathbf{A}\mathbf{y})_{\mathbf{M}} = \mathbf{u}^{\mathsf{T}}\mathbf{A}\mathbf{y} \quad \text{for all } u \in \mathbb{M},$$

where $\langle \cdot, \cdot \rangle_{\mathbb{M}^\star,\mathbb{M}}$ is the duality pairing between $\mathbb{M}$ and its dual. It is now clear that the appropriate norm to measure the residual $\mathbf{b} - \mathbf{A}\mathbf{y}$ is the $\mathbf{M}^{-1}$-norm. Let $\mathscr{A}^\star : \mathbb{M} \to \mathbb{N}^\star$ denote the adjoint operator of $\mathscr{A}$, i.e.,

$$\langle \mathscr{A}^\star u, y \rangle_{\mathbb{N}^\star,\mathbb{N}} := (\mathbf{y}, \mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}}\mathbf{u})_{\mathbf{N}} = \mathbf{y}^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}\mathbf{u}, \quad \text{for all } y \in \mathbb{N}.$$

Finally, let $\mathscr{M} : \mathbb{M} \to \mathbb{M}^\star$, $\mathscr{N} : \mathbb{N} \to \mathbb{N}^\star$ and their inverses $\mathscr{M}^{-1} : \mathbb{M}^\star \to \mathbb{M}$ and $\mathscr{N}^{-1} : \mathbb{N}^\star \to \mathbb{N}$ denote the linear operators whose representations are the matrices $\mathbf{M}$, $\mathbf{N}$ and their inverses.

We define the operator

$$(\mathscr{A}^\star \circ \mathscr{M}^{-1} \circ \mathscr{A}) + \mathscr{N} : \mathbb{N} \to \mathbb{N}^\star$$

as the *normal operator*. This operator appears in the normal equations (2.6). The appropriate norm to measure the residual of the normal equations is the $\mathbf{N}^{-1}$-norm. Similarly, we call the operator

$$(\mathscr{A} \circ \mathscr{N}^{-1} \circ \mathscr{A}^\star) + \mathscr{M} : \mathbb{M} \to \mathbb{M}^\star$$

the *Schur-complement operator*. The residual of the Schur-complement equations is measured in the $\mathbf{M}^{-1}$-norm. The situation is summarized in the commutative diagram of Figure 3.1.

The next sections will regularly refer to Figure 3.1 as they provide appropriate norms in which various quantities such as direct errors $\mathbf{y}_* - \mathbf{y}_k$ and residuals $\mathbf{x}_* - \mathbf{x}_k$, should be measured. The commutative diagram proves to be a consistently useful tool in understanding why such norms are appropriate.

4. Generalized Golub-Kahan Bidiagonalization

Motivated by the least-squares problems of the previous section we recall the standard Golub and Kahan (1965) bidiagonalization process that forms the basis of several numerical methods for such problems. The standard Golub-Kahan bidiagonalization process with initial vector $\mathbf{b} \in \mathbb{R}^n$ can be stated as Algorithm 4.1.

Algorithm 4.1 Golub-Kahan Bidiagonalization

Require: $\bar{\mathbf{A}}$, $\bar{\mathbf{b}}$
1: $\beta_1 \bar{\mathbf{u}}_1 = \bar{\mathbf{b}}$ with $\beta_1 > 0$ so that $\|\bar{\mathbf{u}}_1\|_2 = 1$
2: $\alpha_1 \bar{\mathbf{v}}_1 = \bar{\mathbf{A}}^T \bar{\mathbf{u}}_1$ with $\alpha_1 > 0$ so that $\|\bar{\mathbf{v}}_1\|_2 = 1$
3: **for** $k = 1, 2, \ldots$ **do**
4: $\quad \beta_{k+1} \bar{\mathbf{u}}_{k+1} = \bar{\mathbf{A}} \bar{\mathbf{v}}_k - \alpha_k \bar{\mathbf{u}}_k$ with $\beta_{k+1} > 0$ so that $\|\bar{\mathbf{u}}_{k+1}\|_2 = 1$
5: $\quad \alpha_{k+1} \bar{\mathbf{v}}_{k+1} = \bar{\mathbf{A}}^T \bar{\mathbf{u}}_{k+1} - \beta_{k+1} \bar{\mathbf{v}}_k$ with $\alpha_{k+1} > 0$ so that $\|\bar{\mathbf{v}}_{k+1}\|_2 = 1$.

In exact arithmetic, Algorithm 4.1 generates two sets of orthonormal vectors $\{\bar{\mathbf{u}}_i\}$ and $\{\bar{\mathbf{v}}_i\}$ that can be used to determine the left and right singular vectors of $\bar{\mathbf{A}}$ (Golub and Kahan, 1965).

Consider the application of Algorithm 4.1 to the operator $\bar{\mathbf{A}} = \mathbf{M}^{-\frac{1}{2}} \mathbf{A} \mathbf{N}^{-\frac{1}{2}}$ with initial vector $\bar{\mathbf{b}} = \mathbf{M}^{-\frac{1}{2}} \mathbf{b}$. It is straightforward to verify that after the change of variable $\mathbf{u}_i := \mathbf{M}^{-\frac{1}{2}} \bar{\mathbf{u}}_i$ and $\mathbf{v}_i := \mathbf{N}^{-\frac{1}{2}} \bar{\mathbf{v}}_i$, the resulting process may be written as Algorithm 4.2.

Algorithm 4.2 Generalized Golub-Kahan Bidiagonalization, first variant

Require: $\mathbf{M}$, $\mathbf{A}$, $\mathbf{N}$, $\mathbf{b}$
1: $\beta_1 \mathbf{M} \mathbf{u}_1 = \mathbf{b}$ with $\beta_1 > 0$ so that $\|\mathbf{u}_1\|_{\mathbf{M}} = 1$
2: $\alpha_1 \mathbf{N} \mathbf{v}_1 = \mathbf{A}^T \mathbf{u}_1$ with $\alpha_1 > 0$ so that $\|\mathbf{v}_1\|_{\mathbf{N}} = 1$
3: **for** $k = 1, 2, \ldots$ **do**
4: $\quad \beta_{k+1} \mathbf{M} \mathbf{u}_{k+1} = \mathbf{A} \mathbf{v}_k - \alpha_k \mathbf{M} \mathbf{u}_k$ with $\beta_{k+1} > 0$ so that $\|\mathbf{u}_{k+1}\|_{\mathbf{M}} = 1$
5: $\quad \alpha_{k+1} \mathbf{N} \mathbf{v}_{k+1} = \mathbf{A}^T \mathbf{u}_{k+1} - \beta_{k+1} \mathbf{N} \mathbf{v}_k$ with $\alpha_{k+1} > 0$ so that $\|\mathbf{v}_{k+1}\|_{\mathbf{N}} = 1$.

We refer to Algorithm 4.2 as the *Generalized Golub-Kahan Bidiagonalization* process in the sense that the left and right singular vectors $\{\mathbf{u}_i\}$ and $\{\mathbf{v}_i\}$ are orthonormal with respect to inner products defined by $\mathbf{M}$ and $\mathbf{N}$ respectively. It is important to note that at each iteration, one solve with $\mathbf{M}$ and one solve with $\mathbf{N}$ must be performed. Indeed the terms $\mathbf{M}\mathbf{u}_k$ and $\mathbf{N}\mathbf{v}_k$ in the right-hand sides of the assignments in the loop were computed during the previous pass through the loop. For instance, the computation of β_{k+1} and $\mathbf{u}_{k+1}$ could be detailed as

(1) Set $\hat{\mathbf{u}}_{k+1} = \mathbf{A}\mathbf{v}_k - \alpha_k(\hat{\mathbf{u}}_k/\beta_k)$
(2) Solve $\mathbf{M}\tilde{\mathbf{u}}_{k+1} = \hat{\mathbf{u}}_{k+1}$ for $\tilde{\mathbf{u}}_{k+1}$
(3) Set $\beta_{k+1} = \sqrt{\tilde{\mathbf{u}}_{k+1}^T \hat{\mathbf{u}}_{k+1}}$
(4) Set $\mathbf{u}_{k+1} = \tilde{\mathbf{u}}_{k+1}/\beta_{k+1}$.

The storage per iteration required by Algorithm 4.2 is the same as that required by Algorithm 4.1 with the addition of one n-vector for $\mathbf{M}\mathbf{u}_k$ and one m-vector for $\mathbf{N}\mathbf{v}_k$. The computational effort per iteration is that of Algorithm 4.1 with the addition of one solve with $\mathbf{M}$ and one solve with $\mathbf{N}$.

After k steps of Algorithm 4.2, the situation can be summarized as

$$\mathbf{A}\mathbf{V}_k = \mathbf{M}\mathbf{U}_k\mathbf{B}_k + \beta_{k+1}\mathbf{M}\mathbf{u}_{k+1}\mathbf{e}_k^{\mathsf{T}} \tag{4.1a}$$

$$= \mathbf{M}\mathbf{U}_{k+1}\mathbf{E}_k, \tag{4.1b}$$

$$\mathbf{A}^{\mathsf{T}}\mathbf{U}_{k+1} = \mathbf{N}\mathbf{V}_{k+1}\mathbf{B}_{k+1}^{\mathsf{T}} \tag{4.1c}$$

$$= \mathbf{N}\mathbf{V}_k\mathbf{E}_k^{\mathsf{T}} + \alpha_{k+1}\mathbf{N}\mathbf{v}_{k+1}\mathbf{e}_{k+1}^{\mathsf{T}}, \tag{4.1d}$$

where $\mathbf{e}_k$ is the k-th vector of the canonical basis, $\mathbf{U}_k$ and $\mathbf{V}_k$ are the n-by-k and m-by-k matrices whose columns are $\mathbf{u}_1$ through $\mathbf{u}_k$ and $\mathbf{v}_1$ through $\mathbf{v}_k$, respectively, and

$$\mathbf{B}_k := \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \end{bmatrix}, \quad \mathbf{E}_k := \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{B}_k \\ \beta_{k+1}\mathbf{e}_k^{\mathsf{T}} \end{bmatrix} \tag{4.2}$$

i.e., $\mathbf{B}_k$ is k-by-k lower bidiagonal and $\mathbf{E}_k$ is $\mathbf{B}_k$ with one extra row. The orthogonality properties of the vectors $\mathbf{u}_j$ and $\mathbf{v}_j$ implies that the matrices $\mathbf{M}^{\frac{1}{2}}\mathbf{U}_k$ and $\mathbf{N}^{\frac{1}{2}}\mathbf{V}_k$ are orthogonal for all k.

Algorithm 4.2 is a generalization of the process referred to as GKLB($\mathbf{M}$) by Benbow (1999) and it may be denoted GKLB($\mathbf{M}$,$\mathbf{N}$). An analysis mirroring that of Paige (1974) is instructive in relation to the stopping of the generalized Golub-Kahan process. The first situation that can cause the process to stop is that $\beta_{k+1} = 0$ is generated. In this case, we obtain from (4.1) that, in exact arithmetic,

$$\mathbf{A}\mathbf{V}_k = \mathbf{M}\mathbf{U}_k\mathbf{B}_k, \quad \mathbf{A}^{\mathsf{T}}\mathbf{U}_k = \mathbf{N}\mathbf{V}_k\mathbf{B}_k^{\mathsf{T}}, \quad \mathbf{U}_k^{\mathsf{T}}\mathbf{M}\mathbf{U}_k = \mathbf{I}_k, \quad \mathbf{V}_k^{\mathsf{T}}\mathbf{N}\mathbf{V}_k = \mathbf{I}_k. \tag{4.3}$$

The second situation is that $\alpha_{k+1} = 0$ is generated. In this case, in exact arithmetic,

$$\mathbf{A}^{\mathsf{T}}\mathbf{U}_{k+1} = \mathbf{N}\mathbf{V}_k\mathbf{E}_k^{\mathsf{T}}, \quad \mathbf{A}\mathbf{V}_k = \mathbf{M}\mathbf{U}_{k+1}\mathbf{E}_k, \quad \mathbf{U}_{k+1}^{\mathsf{T}}\mathbf{M}\mathbf{U}_{k+1} = \mathbf{I}_{k+1}, \quad \mathbf{V}_k^{\mathsf{T}}\mathbf{N}\mathbf{V}_k = \mathbf{I}_k. \tag{4.4}$$

Let $\mathbf{E}_k = \mathbf{P}_{k+1}\mathbf{\Sigma}_k\mathbf{Q}_k^{\mathsf{T}}$ be the singular value decomposition of $\mathbf{E}_k$, where $\mathbf{P}_{k+1}$ is orthogonal $(k+1)$-by-$(k+1)$, $\mathbf{\Sigma}_k$ is $(k+1)$-by-k, $\mathbf{Q}_k$ orthogonal is k-by-k and

$$\mathbf{\Sigma}_k = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_k \\ 0 & 0 & \dots & 0 \end{bmatrix}, \qquad \sigma_i > 0,\ i = 1, \dots, k.$$

We have from (4.4) that $\mathbf{A}^{\mathsf{T}}\mathbf{U}_{k+1}\mathbf{P}_{k+1} = \mathbf{N}\mathbf{V}_k\mathbf{Q}_k\mathbf{\Sigma}_k^{\mathsf{T}}$ and $\mathbf{A}\mathbf{V}_k\mathbf{Q}_k = \mathbf{M}\mathbf{U}_{k+1}\mathbf{P}_{k+1}\mathbf{\Sigma}_k$. Equivalently, the last two identities can be stated as

$$(\mathbf{M}^{-\frac{1}{2}}\mathbf{A}\mathbf{N}^{-\frac{1}{2}})^{\mathsf{T}}\bar{\mathbf{U}}_{k+1}\mathbf{P}_{k+1} = \bar{\mathbf{V}}_k\mathbf{Q}_k\mathbf{\Sigma}_k^{\mathsf{T}}, \tag{4.5a}$$

$$(\mathbf{M}^{-\frac{1}{2}}\mathbf{A}\mathbf{N}^{-\frac{1}{2}})\bar{\mathbf{V}}_k\mathbf{Q}_k = \bar{\mathbf{U}}_{k+1}\mathbf{P}_{k+1}\mathbf{\Sigma}_k. \tag{4.5b}$$

where we used $\bar{\mathbf{u}}_j = \mathbf{M}^{\frac{1}{2}}\mathbf{u}_j$ and $\bar{\mathbf{v}}_j = \mathbf{N}^{\frac{1}{2}}\mathbf{v}_j$ for all j. The relations (4.5) show that $\bar{\mathbf{A}} := \mathbf{M}^{-\frac{1}{2}}\mathbf{A}\mathbf{N}^{-\frac{1}{2}}$ has a zero singular value and that the associated right singular

vector is the last column of $\bar{\mathbf{U}}_{k+1}\mathbf{P}_{k+1}$. This last column can be written

$$\mathbf{t} = \sum_{j=1}^{k+1} p_{k+1,j}\mathbf{M}^{\frac{1}{2}}\mathbf{u}_j,$$

where $\mathbf{p}_{k+1}$ is the last column of $\mathbf{P}_{k+1}$. This vector $\mathbf{t}$ is a linear combination of the vectors $\mathbf{M}^{\frac{1}{2}}\mathbf{u}_j$ and must lie in $\text{Null}(\bar{\mathbf{A}}^{\mathsf{T}}) = \text{Null}(\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-\frac{1}{2}})$. Assume now that $\mathbf{b} \in \text{Range}(\mathbf{A})$. Then $\mathbf{u}_1 \in \text{Range}(\mathbf{M}^{-1}\mathbf{A})$. A recursion argument easily establishes that each $\mathbf{u}_j \in \text{Range}(\mathbf{M}^{-1}\mathbf{A})$. In this case, the vector $\mathbf{t}$ thus lies in $\text{Range}(\mathbf{M}^{-\frac{1}{2}}\mathbf{A})$ which is in contradiction with the previous conclusion that $\mathbf{t} \in \text{Null}(\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-\frac{1}{2}})$. Therefore, if $\mathbf{b} \in \text{Range}(\mathbf{A})$, Algorithm 4.2 cannot terminate with $\alpha_{k+1} = 0$. It must thus terminate with $\beta_{k+1} = 0$ and the final situation must be described by (4.3).

Conversely, if (4.3) describes the final situation, necessarily $\mathbf{u}_1 \in \text{Range}(\mathbf{M}^{-1}\mathbf{A})$, i.e., $\mathbf{b} \in \text{Range}(\mathbf{A})$ because $\mathbf{U}_k = \mathbf{M}^{-1}\mathbf{A}\mathbf{V}_k\mathbf{B}_k^{-1}$, which shows that all $\mathbf{u}_j$ lie in $\text{Range}(\mathbf{M}^{-1}\mathbf{A})$. Therefore if $\mathbf{b} \notin \text{Range}(\mathbf{A})$, the final situation must be described by (4.4).

In the same way that Algorithm 4.1 is closely related to the singular-value decomposition of $\mathbf{A}$, Algorithm 4.2 is related to the elliptic singular-value decomposition of $\mathbf{A}$ (Arioli, 2010) in the sense that the families $\{\mathbf{u}_i\}$ and $\{\mathbf{v}_i\}$ determine families $\{\mathbf{w}_i\}$ and $\{\mathbf{z}_i\}$ satisfying

$$\begin{aligned} \mathbf{A}\mathbf{z}_i &= \sigma_i\mathbf{M}\mathbf{w}_i, & \mathbf{z}_i^{\mathsf{T}}\mathbf{N}\mathbf{z}_j &= \delta_{ij}, \\ \mathbf{A}^{\mathsf{T}}\mathbf{w}_i &= \sigma_i\mathbf{N}\mathbf{z}_i, & \mathbf{w}_i^{\mathsf{T}}\mathbf{M}\mathbf{w}_j &= \delta_{ij}. \end{aligned}$$

In matrix form, this can also be cast as the generalized eigenvalue problem

$$\begin{bmatrix} & \mathbf{A} \\ \mathbf{A}^{\mathsf{T}} & \end{bmatrix}\begin{bmatrix} \mathbf{w} \\ \mathbf{z} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{M} & \\ & \mathbf{N} \end{bmatrix}\begin{bmatrix} \mathbf{w} \\ \mathbf{z} \end{bmatrix}.$$

The positive scalars σ_i are referred to as the elliptic singular values of $\mathbf{A}$ and the families $\{\mathbf{w}_i\}$ and $\{\mathbf{z}_i\}$ are its left and right elliptic singular vectors, respectively. The latter may equivalently be interpreted as the stationary points of the indefinite quadratic mapping

$$(\mathbf{w}, \mathbf{z}) \mapsto \mathbf{w}^{\mathsf{T}}\mathbf{A}\mathbf{z}$$

restricted to the unit sphere $\|\mathbf{w}\|_{\mathbf{M}} = 1$ and $\|\mathbf{z}\|_{\mathbf{N}} = 1$. In exact arithmetic, we have from (4.1) that $\bar{\mathbf{U}}_k^{\mathsf{T}}(\mathbf{M}^{-\frac{1}{2}}\mathbf{A}\mathbf{N}^{-\frac{1}{2}})\bar{\mathbf{V}}_k = \mathbf{B}_k$, where $\bar{\mathbf{u}}_j = \mathbf{M}^{\frac{1}{2}}\mathbf{u}_j$ and $\bar{\mathbf{v}}_j = \mathbf{N}^{\frac{1}{2}}\mathbf{v}_j$. Since the matrices $\bar{\mathbf{U}}_k$ and $\bar{\mathbf{V}}_k$ are orthogonal for all k, the singular values of $\mathbf{B}_{\min(m,n)}$ are the same as those of $\mathbf{M}^{-\frac{1}{2}}\mathbf{A}\mathbf{N}^{-\frac{1}{2}}$.

There are algebraically equivalent alternatives to Algorithm 4.2. For instance, if we use instead the change of variables $\mathbf{u}_i := \mathbf{M}^{\frac{1}{2}}\bar{\mathbf{u}}_i$ and $\mathbf{v}_i := \mathbf{N}^{-\frac{1}{2}}\bar{\mathbf{v}}_i$, and use the initial vector $\mathbf{M}^{\frac{1}{2}}\mathbf{b}$, we obtain the process described in Algorithm 4.3.

The process of Algorithm 4.3, which we could denote GKLB($\mathbf{M}^{-1}$, $\mathbf{N}$) generalizes the process referred to by Benbow (1999) as GKLB($\mathbf{M}^{-1}$). As both are mathematically equivalent, in the rest of this paper, we concentrate on the process GKLB($\mathbf{M}$,$\mathbf{N}$) described by Algorithm 4.2. However, all methods examined in the next sections could be examined instead with Algorithm 4.3 and similar conclusions could be drawn.

Algorithm 4.3 Generalized Golub-Kahan Bidiagonalization, second variant

Require: $\mathbf{M}$, $\mathbf{A}$, $\mathbf{N}$, $\mathbf{b}$

$\beta_1 \mathbf{M}^{-1}\mathbf{u}_1 = \mathbf{b}$ with $\beta_1 > 0$ so that $\|\mathbf{u}_1\|_{\mathbf{M}^{-1}} = 1$

$\alpha_1 \mathbf{N}\mathbf{v}_1 = \mathbf{A}^T\mathbf{M}^{-1}\mathbf{u}_1$ with $\alpha_1 > 0$ so that $\|\mathbf{v}_1\|_{\mathbf{N}} = 1$

for $k = 1, 2, \ldots$ **do**

  $\beta_{k+1}\mathbf{u}_{k+1} = \mathbf{A}\mathbf{v}_k - \alpha_k\mathbf{u}_k$ with $\beta_{k+1} > 0$ so that $\|\mathbf{u}_{k+1}\|_{\mathbf{M}^{-1}} = 1$

  $\alpha_{k+1}\mathbf{N}\mathbf{v}_{k+1} = \mathbf{A}^T\mathbf{M}^{-1}\mathbf{u}_{k+1} - \beta_{k+1}\mathbf{N}\mathbf{v}_k$ with $\alpha_{k+1} > 0$ so that $\|\mathbf{v}_{k+1}\|_{\mathbf{N}} = 1$.

5. Properties of SQD matrices and of their Krylov spaces

5.1. **Eigenvalues.** From Sylvester's law of inertia, the congruence relation

$$\begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & -\mathbf{N} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_n & \\ \mathbf{A}^T\mathbf{M}^{-1} & \mathbf{I}_m \end{bmatrix} \begin{bmatrix} \mathbf{M} & \\ & -(\mathbf{A}^T\mathbf{M}^{-1}\mathbf{A} + \mathbf{N}) \end{bmatrix} \begin{bmatrix} \mathbf{I}_n & \mathbf{M}^{-1}\mathbf{A} \\ & \mathbf{I}_m \end{bmatrix} \tag{5.1}$$

shows that the coefficient matrix of (1.1) always possesses precisely n positive and m negative eigenvalues. Note that a second possible decomposition illustrating this result is

$$\begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & -\mathbf{N} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_n & -\mathbf{A}\mathbf{N}^{-1} \\ & \mathbf{I}_m \end{bmatrix} \begin{bmatrix} \mathbf{A}\mathbf{N}^{-1}\mathbf{A}^T + \mathbf{M} & \\ & -\mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{I}_n & \\ -\mathbf{N}^{-1}\mathbf{A}^T & \mathbf{I}_m \end{bmatrix}. \tag{5.2}$$

The result below is more precise. Consider centered preconditioning of (1.1) with $\mathbf{f} = \mathbf{b}$ and $\mathbf{g} = \mathbf{0}$:

$$\begin{bmatrix} \mathbf{M}^{-\frac{1}{2}} & \\ & \mathbf{N}^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & -\mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{M}^{-\frac{1}{2}} & \\ & \mathbf{N}^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{M}^{\frac{1}{2}}\mathbf{x} \\ \mathbf{N}^{\frac{1}{2}}\mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{M}^{-\frac{1}{2}}\mathbf{b} \\ \mathbf{0} \end{bmatrix}. \tag{5.3}$$

It is straightforward to verify that the coefficient matrix of the previous system is

$$\bar{\mathbf{K}} := \begin{bmatrix} \mathbf{I}_n & \bar{\mathbf{A}} \\ \bar{\mathbf{A}}^T & -\mathbf{I}_m \end{bmatrix} \quad \text{with} \quad \bar{\mathbf{A}} := \mathbf{M}^{-\frac{1}{2}}\mathbf{A}\mathbf{N}^{-\frac{1}{2}}. \tag{5.4}$$

The next result gives the eigenvalues of $\bar{\mathbf{K}}$. It is a special case of (Saunders, 1995, Result 2).

Theorem 5.1. *Suppose $\bar{\mathbf{A}}$ has rank $p \leq \min(m, n)$ with nonzero singular values $\sigma_1, \ldots, \sigma_p$. The eigenvalues of $\bar{\mathbf{K}}$ are*

- $\lambda = +1$ *with multiplicity* $n - p$,
- $\lambda = -1$ *with multiplicity* $m - p$,
- $\lambda = \pm\sqrt{1 + \sigma_k^2}$, $k = 1, \ldots, p$.

The scalars σ_k are the singular values of $\mathbf{M}^{-\frac{1}{2}}\mathbf{A}\mathbf{N}^{-\frac{1}{2}}$, which we call the *elliptic singular values* of $\mathbf{A}$. A similar result clearly also holds if we replace $\mathbf{M}^{\frac{1}{2}}$ with $\mathbf{L}$ and $\mathbf{N}^{\frac{1}{2}}$ with $\mathbf{R}$, where $\mathbf{M} = \mathbf{L}\mathbf{L}^T$ and $\mathbf{N} = \mathbf{R}^T\mathbf{R}$.

Theorem 5.1 implies that the spectrum of $\bar{\mathbf{K}}$ is symmetric, i.e., if λ is an eigenvalue of $\bar{\mathbf{K}}$, then $-\lambda$ is another. An important result related to operators with symmetric spectrum due to Fischer (2011, Theorem 6.9.9) will prove to be instrumental to our analysis.

Consider $\mathbf{K}$, $\mathbf{H}$ and $\bar{\mathbf{K}}$ as defined in (3.5), (3.6) and (5.4), and observe that

$$\mathbf{K} = \mathbf{H}^{\frac{1}{2}} \bar{\mathbf{K}} \mathbf{H}^{\frac{1}{2}}. \tag{5.5}$$

By direct computation,

$$\bar{\mathbf{K}}^2 = \begin{bmatrix} \mathbf{I}_n + \bar{\mathbf{A}}\bar{\mathbf{A}}^{\mathsf{T}} & \\ & \mathbf{I}_m + \bar{\mathbf{A}}^{\mathsf{T}}\bar{\mathbf{A}} \end{bmatrix} := \bar{\mathbf{D}}. \tag{5.6}$$

From (5.6) and the symmetry of $\bar{\mathbf{K}}$, we have the following properties:

$$\bar{\mathbf{K}}^{-1} = \bar{\mathbf{D}}^{-1}\bar{\mathbf{K}} = \bar{\mathbf{K}}\bar{\mathbf{D}}^{-1} \tag{5.7}$$

$$\bar{\mathbf{K}}\bar{\mathbf{D}} = \bar{\mathbf{K}}^3 = \bar{\mathbf{D}}\bar{\mathbf{K}} \tag{5.8}$$

$$\mathbf{K}\mathbf{H}^{-1}\mathbf{K} = \mathbf{H}^{\frac{1}{2}}\bar{\mathbf{D}}\mathbf{H}^{\frac{1}{2}} = \begin{bmatrix} \mathbf{M} + \mathbf{A}\mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}} & \\ & \mathbf{N} + \mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A} \end{bmatrix} := \mathbf{D}. \tag{5.9}$$

Note that $\mathbf{D}$ contains the coefficient matrix of both the normal equations (2.6) and the Schur-complement equations (2.7). Finally, from (5.9), we have

$$\mathbf{K}^{-1} = \mathbf{D}^{-1}\mathbf{K}\mathbf{H}^{-1} = \mathbf{H}^{-1}\mathbf{K}\mathbf{D}^{-1}. \tag{5.10}$$

Because (5.8) says that $\bar{\mathbf{D}}$ and $\bar{\mathbf{K}}$ commute, both matrices can be simultaneously diagonalized. In addition, (5.5) and (5.9) imply that $\mathbf{D}$ and $\mathbf{K}$ can be simultaneously diagonalized by the solutions of the generalized eigenvalue problem

$$\mathbf{K}\mathbf{z} = \lambda_j \mathbf{H}\mathbf{z},$$

where the λ_j, $j = 1, \ldots, p = \operatorname{rank}(\bar{\mathbf{A}})$ are the same eigenvalues presented in Theorem 5.1. Again, the entire discussion above remains if we replace $\mathbf{H}^{\frac{1}{2}}$ by the Cholesky factor of $\mathbf{H}$.

5.2. **Krylov subspaces.** We denote

$$\mathcal{K}_i(\bar{\mathbf{K}}, \bar{\mathbf{z}}) = \text{Range}\left\{\bar{\mathbf{K}}^i\bar{\mathbf{z}}, \bar{\mathbf{K}}^{i-1}\bar{\mathbf{z}}, \ldots, \bar{\mathbf{K}}\bar{\mathbf{z}}, \bar{\mathbf{z}}\right\} \tag{5.11}$$

the i-th Krylov subspace generated by $\bar{\mathbf{K}}$ and a vector $\bar{\mathbf{z}}$. Note that $\mathcal{K}_i(\bar{\mathbf{K}}, \bar{\mathbf{z}})$ is also the i-th Krylov subspace generated by $\mathbf{K}$ symmetrically preconditioned by $\mathbf{H}^{\frac{1}{2}}$ and the vector $\mathbf{z} = \mathbf{H}^{\frac{1}{2}}\bar{\mathbf{z}}$. Moreover, taking into account (3.4), the discussion in §3.2 and (5.5), we have

$$\mathbf{H}^{-1}\mathbf{K} = \mathbf{H}^{-\frac{1}{2}}\bar{\mathbf{K}}\mathbf{H}^{\frac{1}{2}}, \tag{5.12}$$

and

$$\mathcal{K}_i(\mathbf{H}^{-1}\mathbf{K}, \mathbf{w}) = \mathbf{H}^{-\frac{1}{2}}\mathcal{K}_i(\bar{\mathbf{K}}, \mathbf{z}), \qquad \text{where} \qquad \mathbf{w} = \mathbf{H}^{-\frac{1}{2}}\mathbf{z}. \tag{5.13}$$

Owing to the symmetry of the spectrum of $\bar{\mathbf{K}}$—see Theorem 5.1—it is known (Fischer, 2011, Theorem 6.9.9) and (Freund et al., 1991) that Lanczos-based algorithms such as MINRES perform redundant iterations. Taking into account the structure of $\bar{\mathbf{K}}$, it is possible to be more precise.

From (5.6) and (5.8), we have, for all $k \geq 0$,

$$\bar{\mathbf{K}}^{2k} = \bar{\mathbf{D}}^k \qquad \text{and} \qquad \bar{\mathbf{K}}^{2k+1} = \bar{\mathbf{K}}\bar{\mathbf{D}}^k = \bar{\mathbf{D}}^k\bar{\mathbf{K}}. \tag{5.14}$$

Consequently, for all $i \geq 0$, the Krylov subspace $\mathcal{K}_i(\bar{\mathbf{K}}, \bar{\mathbf{z}})$ can be written as the direct sum

$$\begin{aligned} \mathcal{K}_i(\bar{\mathbf{K}}, \bar{\mathbf{z}}) &= \mathcal{K}_{\lfloor i/2 \rfloor}(\bar{\mathbf{D}}, \bar{\mathbf{z}}) \oplus \mathcal{K}_{\lfloor i/2 \rfloor}(\bar{\mathbf{D}}, \bar{\mathbf{K}}\bar{\mathbf{z}}) \\ &= \mathcal{K}_{\lfloor i/2 \rfloor}(\bar{\mathbf{D}}, \bar{\mathbf{z}}) \oplus \bar{\mathbf{K}}\mathcal{K}_{\lfloor i/2 \rfloor}(\bar{\mathbf{D}}, \bar{\mathbf{z}}). \end{aligned} \tag{5.15}$$

Let $\bar{\mathbf{D}}_1$ and $\bar{\mathbf{D}}_2$ be defined such that $\bar{\mathbf{D}} = \text{blkdiag}(\bar{\mathbf{D}}_1, \bar{\mathbf{D}}_2)$. Then, if $\bar{z} = (\bar{z}_1, \bar{z}_2)$, we have

$$\mathcal{K}_i(\bar{\mathbf{D}}, \bar{\mathbf{z}}) = \begin{bmatrix} \mathcal{K}_i(\bar{\mathbf{D}}_1, \bar{\mathbf{z}}_1) \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ \mathcal{K}_i(\bar{\mathbf{D}}_2, \bar{\mathbf{z}}_2) \end{bmatrix} \tag{5.16}$$

and

$$\bar{\mathbf{K}}\mathcal{K}_i(\bar{\mathbf{D}}, \bar{\mathbf{z}}) = \begin{bmatrix} \mathcal{K}_i(\bar{\mathbf{D}}_1, \bar{\mathbf{z}}_1) \\ \bar{\mathbf{A}}^{\mathsf{T}}\mathcal{K}_i(\bar{\mathbf{D}}_1, \bar{\mathbf{z}}_1) \end{bmatrix} \oplus \begin{bmatrix} \bar{\mathbf{A}}\mathcal{K}_i(\bar{\mathbf{D}}_2, \bar{\mathbf{z}}_2) \\ -\mathcal{K}_i(\bar{\mathbf{D}}_2, \bar{\mathbf{z}}_2) \end{bmatrix} \tag{5.17}$$

$$= \begin{bmatrix} \mathcal{K}_i(\bar{\mathbf{D}}_1, \bar{\mathbf{z}}_1) \\ \mathcal{K}_i(\bar{\mathbf{D}}_2, \bar{\mathbf{A}}^{\mathsf{T}}\bar{\mathbf{z}}_1) \end{bmatrix} \oplus \begin{bmatrix} \mathcal{K}_i(\bar{\mathbf{D}}_1, \bar{\mathbf{A}}\bar{\mathbf{z}}_2) \\ -\mathcal{K}_i(\bar{\mathbf{D}}_2, \bar{\mathbf{z}}_2) \end{bmatrix}. \tag{5.18}$$

In particular, if we choose $\mathbf{z} = (\mathbf{b}, \mathbf{0})$ or $(\mathbf{0}, \mathbf{g})$, we have, respectively,

$$\mathcal{K}_i\left(\bar{\mathbf{D}}, \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}\right) = \begin{bmatrix} \mathcal{K}_i(\bar{\mathbf{D}}_1, \mathbf{b}) \\ 0 \end{bmatrix} \qquad \text{and} \qquad \mathcal{K}_i\left(\bar{\mathbf{D}}, \begin{bmatrix} \mathbf{0} \\ \mathbf{g} \end{bmatrix}\right) = \begin{bmatrix} 0 \\ \mathcal{K}_i(\bar{\mathbf{D}}_2, \mathbf{g}) \end{bmatrix}.$$

With $\mathbf{z} = (\mathbf{b}, \mathbf{0})$, we finally obtain

$$\mathcal{K}_i\left(\bar{\mathbf{K}}, \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}\right) = \begin{bmatrix} \mathcal{K}_{\lfloor i/2 \rfloor}(\bar{\mathbf{D}}_1, \mathbf{b}) \\ \mathbf{0} \end{bmatrix} \oplus \begin{bmatrix} \mathbf{0} \\ \mathcal{K}_{\lfloor i/2 \rfloor + 1}(\bar{\mathbf{D}}_2, \mathbf{A}^{\mathsf{T}}\mathbf{b}) \end{bmatrix}.$$

6. Methods Based on Reduced Equations

This section presents a family of four methods based on the normal and Schur-complement equations. Three methods are generalizations of known methods in appropriate metrics: LSQR, CRAIG and LSMR. The last one is new and may be viewed as an alternative to LSMR when $m < n$. It also serves as an essential tool to explain the behavior of MINRES on (1.1) in §8.4. For each method, we give implementation details in order to be complete and to provide a self-contained reference. The implementation details of LSQR and LSMR are our interpretation of the descriptions in (Paige and Saunders, 1982) and (Fong and Saunders, 2011). The implementation details of CRAIG were pieced together from various hints scattered across the literature and those of CRAIG-MR are new although they essentially mirror LSMR.

6.1. Generalized LSQR.

The generalized LSQR method seeks a solution to the normal equations (2.6). At the end of the k-th iteration of Algorithm 4.2, we may seek an approximation $\mathbf{y}_k$ to the solution $\mathbf{y}$ of (2.6) in the k-th Krylov subspace spanned by $\{v_1, \ldots, v_k\}$, i.e.,

$$\mathbf{y} \approx \mathbf{y}_k := \mathbf{V}_k \bar{\mathbf{y}}_k$$

for some $\bar{\mathbf{y}}_k \in \mathbb{R}^k$. Using (4.1), we have, in exact arithmetic, $\mathbf{A}\mathbf{y}_k = \mathbf{A}\mathbf{V}_k\bar{\mathbf{y}}_k = \mathbf{M}\mathbf{U}_{k+1}\mathbf{E}_k\bar{\mathbf{y}}_k$ so that

$$\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A}\mathbf{y}_k = (\mathbf{N}\mathbf{V}_k\mathbf{E}_k^{\mathsf{T}} + \alpha_{k+1}\mathbf{N}\mathbf{v}_{k+1}\mathbf{e}_{k+1}^{\mathsf{T}})\mathbf{E}_k\bar{\mathbf{y}}_k,$$

and, using the definition of $\mathbf{y}_k$, $\mathbf{N}\mathbf{y}_k = \mathbf{N}V_k\bar{\mathbf{y}}_k$. Similarly, the initialization of Algorithm 4.2 guarantees that $\mathbf{b} = \mathbf{M}\mathbf{U}_{k+1}(\beta_1\mathbf{e}_1)$ holds to machine precision at each iteration. This implies that

$$\mathbf{A}^T\mathbf{M}^{-1}\mathbf{b} = (\mathbf{N}\mathbf{V}_k\mathbf{E}_k^T + \alpha_{k+1}\mathbf{N}\mathbf{v}_{k+1}\mathbf{e}_{k+1}^T)(\beta_1\mathbf{e}_1).$$

Finally, (2.6) may be equivalently written

$$((\mathbf{N}\mathbf{V}_k\mathbf{E}_k^T + \alpha_{k+1}\mathbf{N}\mathbf{v}_{k+1}\mathbf{e}_{k+1}^T)\mathbf{E}_k + \mathbf{N}\mathbf{V}_k)\bar{\mathbf{y}}_k = (\mathbf{N}\mathbf{V}_k\mathbf{E}_k^T + \alpha_{k+1}\mathbf{N}\mathbf{v}_{k+1}\mathbf{e}_{k+1}^T)(\beta_1\mathbf{e}_1).$$

Upon premultiplying the previous equality with $\mathbf{V}_k^T$ and using the fact that the $\mathbf{v}_k$'s are $\mathbf{N}$-orthonormal in exact arithmetic, we obtain

$$(\mathbf{E}_k^T\mathbf{E}_k + \mathbf{I}_k)\bar{\mathbf{y}}_k = \mathbf{E}_k^T(\beta_1\mathbf{e}_1), \tag{6.1}$$

which are the optimality conditions of the linear least-squares problem

$$\underset{\bar{\mathbf{y}}\in\mathbb{R}^k}{\text{minimize}} \ \tfrac{1}{2}\left\| \begin{bmatrix}\mathbf{E}_k \\ \mathbf{I}_k\end{bmatrix}\bar{\mathbf{y}} - \begin{bmatrix}\beta_1\mathbf{e}_1 \\ 0\end{bmatrix}\right\|_2^2. \tag{6.2}$$

The latter is exactly the k-th regularized least-squares subproblem solved by LSQR with regularization parameter $\lambda = 1$. Equivalently, $\bar{\mathbf{y}}_k$ solves the SQD subsystem

$$\begin{bmatrix}\mathbf{I}_{k+1} & \mathbf{E}_k \\ \mathbf{E}_k^T & -\mathbf{I}_k\end{bmatrix}\begin{bmatrix}\bar{\mathbf{x}}_k \\ \bar{\mathbf{y}}_k\end{bmatrix} = \begin{bmatrix}\beta_1\mathbf{e}_1 \\ \mathbf{0}\end{bmatrix} \tag{6.3}$$

for some $\bar{\mathbf{x}}_k$. As in (Benbow, 1999), this means that all that need be changed in LSQR to solve (1.1) with $\mathbf{f} = \mathbf{b}$ and $\mathbf{g} = \mathbf{0}$ is Algorithm 4.1, which should be replaced with Algorithm 4.2.

The ℓ_2-norm residual satisifes

$$\|\bar{\mathbf{x}}_k\|_2 = \|\mathbf{M}^{\frac{1}{2}}\mathbf{U}_{k+1}\bar{\mathbf{x}}_k\|_2 = \|\mathbf{M}^{-\frac{1}{2}}(\mathbf{M}\mathbf{x}_k)\|_2 = \|\mathbf{M}\mathbf{x}_k\|_{\mathbf{M}^{-1}} = \|\mathbf{x}_k\|_{\mathbf{M}},$$

where we used the fact that $\mathbf{M}^{\frac{1}{2}}\mathbf{U}_{k+1}$ is an orthogonal matrix. Thus $\|\mathbf{M}\mathbf{x}_k\|_{\mathbf{M}^{-1}}$ may be updated recursively by updating $\|\bar{\mathbf{x}}_k\|_2$ as in the original LSQR, and the sequence $\{\|\mathbf{M}\mathbf{x}_k\|_{\mathbf{M}^{-1}}\}$ is non-increasing.

It is convenient to solve (6.1) by computing a $(2k+1)$-by-$(2k+1)$ orthogonal matrix $\mathbf{Q}_k$ as a product of Givens rotations and a $(2k+1)$-by-k upper bidiagonal matrix $\mathbf{R}_k$ such that

$$\tilde{\mathbf{E}}_k := \begin{bmatrix}\mathbf{E}_k \\ \mathbf{I}_k\end{bmatrix} = \mathbf{Q}_k\mathbf{R}_k. \tag{6.4}$$

We give the details of the construction of $\mathbf{Q}_k$ in the next section.

The following result is algebraic and generalizes (Saunders, 1995, Result 8). It is based on the observation that the coefficient matrix of (2.6) may also be written

$$\mathbf{N}^{\frac{1}{2}}\left((\mathbf{M}^{-\frac{1}{2}}\mathbf{A}\mathbf{N}^{-\frac{1}{2}})^T(\mathbf{M}^{-\frac{1}{2}}\mathbf{A}\mathbf{N}^{-\frac{1}{2}}) + \mathbf{I}_n\right)\mathbf{N}^{\frac{1}{2}}.$$

Theorem 6.1. *The generalized* LSQR *iterates on (2.4) are the same as those generated by the standard conjugate gradient method on the positive definite system (2.6) with preconditioner* $\mathbf{N}$.

Proof. Proceeding as above, using (4.1) and post-multiplying the coefficient matrix of (2.6) by $\mathbf{V}_k$, we have

$$\begin{aligned}(\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A}+\mathbf{N})\mathbf{V}_k &= \mathbf{N}(\mathbf{V}_k\mathbf{E}_k^{\mathsf{T}}\mathbf{E}_k+\alpha_{k+1}\mathbf{v}_{k+1}\mathbf{e}_{k+1}^{\mathsf{T}}\mathbf{E}_k+\mathbf{V}_k)\\ &= \mathbf{N}(\mathbf{V}_k(\mathbf{E}_k^{\mathsf{T}}\mathbf{E}_k+\mathbf{I}_k)+\alpha_{k+1}\beta_{k+1}\mathbf{v}_{k+1}\mathbf{e}_k^{\mathsf{T}}),\end{aligned}$$

where we used the fact that $\mathbf{e}_{k+1}^{\mathsf{T}}\mathbf{E}_k = \beta_{k+1}\mathbf{e}_k^{\mathsf{T}}$. The matrix $\mathbf{T}_k := \mathbf{E}_k^{\mathsf{T}}\mathbf{E}_k + \mathbf{I}_k$ is tridiagonal, symmetric and positive definite, and its i-th off-diagonal element is $\alpha_i\beta_i$. Therefore, upon comparing with (3.2), the above represents a Lanczos process applied to the coefficient matrix of (2.6) in a metric defined by $\mathbf{N}$. Moreover, by definition of $\mathbf{Q}_k$ and $\mathbf{R}_k$,

$$\mathbf{E}_k^{\mathsf{T}}\mathbf{E}_k+\mathbf{I}_k = \begin{bmatrix}\mathbf{E}_k^{\mathsf{T}} & \mathbf{I}_k\end{bmatrix}\mathbf{Q}_k\mathbf{Q}_k^{\mathsf{T}}\begin{bmatrix}\mathbf{E}_k\\ \mathbf{I}_k\end{bmatrix} = \mathbf{R}_k^{\mathsf{T}}\mathbf{R}_k.$$

Therefore, $\mathbf{R}_k$ is the Cholesky factor of $\mathbf{T}_k$, updated at each iteration, and the generalized LSQR method is equivalent to the method of conjugate gradients applied to (2.6) with preconditioner $\mathbf{N}$. □

6.2. **Generalized LSQR Recursive Expressions.** In this section we give update formulae to perform the factorization (6.4) iteratively. Substituting the identity

$$\mathbf{R}_k^{\mathsf{T}}\mathbf{R}_k\bar{\mathbf{y}}_k = \begin{bmatrix}\mathbf{E}_k^{\mathsf{T}} & \mathbf{I}_k\end{bmatrix}\begin{bmatrix}\mathbf{E}_k\\ \mathbf{I}_k\end{bmatrix}\bar{\mathbf{y}}_k = (\mathbf{E}_k^{\mathsf{T}}\mathbf{E}_k+\mathbf{I}_k)\bar{\mathbf{y}}_k,$$

into (6.1), we obtain

$$\mathbf{R}_k^{\mathsf{T}}\mathbf{R}_k\bar{\mathbf{y}}_k = \mathbf{E}_k^{\mathsf{T}}\beta_1\mathbf{e}_1 = \begin{bmatrix}\mathbf{E}_k^{\mathsf{T}} & \mathbf{I}_k\end{bmatrix}\begin{bmatrix}\beta_1\mathbf{e}_1\\ \mathbf{0}\end{bmatrix} = \mathbf{R}_k^{\mathsf{T}}\mathbf{Q}_k^{\mathsf{T}}\begin{bmatrix}\beta_1\mathbf{e}_1\\ \mathbf{0}\end{bmatrix} = \alpha_1\beta_1\mathbf{e}_1. \tag{6.5}$$

The $(2j+1, j)$-th element of $\tilde{\mathbf{E}}_k$ (equal to 1) may be zeroed out by applying a Givens rotation acting on rows j and $2j$, denoted $\mathbf{Q}_{j,2j+1}^{\mathsf{T}}$, the last index indicating the row of the element being zeroed out. This rotation does not create any new nonzero in the other columns of $\tilde{\mathbf{E}}_k$. Its effect may be represented schematically as (ignoring irrelevant rows and columns)

$$\begin{array}{c} \\ j \\ 2j+1 \end{array}\!\!\begin{array}{c}\begin{array}{cc} j & 2j+1 \end{array}\\ \begin{bmatrix} c_j & s_j \\ s_j & -c_j \end{bmatrix}\end{array}\!\!\begin{array}{c}\begin{array}{cc} j & j+1 \end{array}\\ \begin{bmatrix} \hat{\alpha}_j & 0 \\ 1 & 0 \end{bmatrix}\end{array} = \begin{array}{c}\begin{array}{cc} j & j+1 \end{array}\\ \begin{bmatrix} \hat{\rho}_j & 0 \\ 0 & 0 \end{bmatrix}\end{array},$$

where $\hat{\rho}_j := \sqrt{\hat{\alpha}_j^2+1}$, $c_j := \hat{\alpha}_j/\hat{\rho}_j$, $s_j = 1/\hat{\rho}_j$ and initially, $\hat{\alpha}_1 = \alpha_1$. Next, the β_{j+1} in position $(j+1, j)$ may be zeroed out by a second Givens rotation acting on rows j and $j+1$, denoted $\mathbf{Q}_{j,j+1}^{\mathsf{T}}$. This rotation creates a new nonzero element in position $(j, j+1)$ as the following schema illustrates

$$\begin{array}{c} \\ j \\ j+1 \end{array}\!\!\begin{array}{c}\begin{array}{cc} j & j+1 \end{array}\\ \begin{bmatrix} \bar{c}_j & \bar{s}_j \\ \bar{s}_j & -\bar{c}_j \end{bmatrix}\end{array}\!\!\begin{array}{c}\begin{array}{cc} j & j+1 \end{array}\\ \begin{bmatrix} \hat{\rho}_j & 0 \\ \beta_{j+1} & \alpha_{j+1} \end{bmatrix}\end{array} = \begin{array}{c}\begin{array}{cc} j & j+1 \end{array}\\ \begin{bmatrix} \rho_j & \theta_{j+1} \\ 0 & \hat{\alpha}_{j+1} \end{bmatrix}\end{array},$$

where $\rho_j := \sqrt{\hat{\rho}_j^2+\beta_{j+1}^2}$, $\bar{c}_j := \hat{\rho}_j/\rho_j$, $\bar{s}_j := \beta_{j+1}/\rho_j$, $\theta_{j+1} := \bar{s}_j\alpha_{j+1}$ and $\hat{\alpha}_{j+1} := -\bar{c}_j\alpha_{j+1}$. It is now easy to see that the overall orthogonal matrix $\mathbf{Q}_k$ is given by

$$\mathbf{Q}_k = (\mathbf{Q}_{1,4}\mathbf{Q}_{1,2})(\mathbf{Q}_{2,6}\mathbf{Q}_{2,3})\cdots(\mathbf{Q}_{k,2k}\mathbf{Q}_{k,k+1}).$$

Recalling that $\mathbf{E}_k$ is $(k+1)$-by-k, the result of the first k Givens rotations may be described as

$$\mathbf{Q}_k^T \begin{bmatrix} \mathbf{E}_k & \beta_1 \mathbf{e}_1 \\ \mathbf{I}_k & \mathbf{0} \end{bmatrix} = \mathbf{Q}_k^T \begin{bmatrix} \mathbf{B}_k & \beta_1 \mathbf{e}_1 \\ \beta_{k+1} \mathbf{e}_k^T & 0 \\ \mathbf{I}_k & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_k & \mathbf{z}_k \\ \mathbf{0} & \bar{\zeta}_{k+1} \\ \mathbf{0} & \mathbf{w}_k \end{bmatrix},$$

where $\mathbf{z}_k = (\zeta_1, \ldots, \zeta_k)$, $\mathbf{w}_k = (\omega_1, \ldots, \omega_k)$ and $\bar{\zeta}_{k+1}$ will be updated into ζ_{k+1} with the next Givens rotation.

The update of the right-hand side may be visualized as

$$\begin{array}{c} \\ j \\ j+1 \\ 2j+1 \end{array} \begin{array}{c} \begin{array}{ccc} j & j+1 & 2j+1 \end{array} \\ \begin{bmatrix} \bar{c}_j & \bar{s}_j & \\ \bar{s}_j & -\bar{c}_j & \\ & & 1 \end{bmatrix} \end{array} \begin{array}{c} \begin{array}{ccc} j & j+1 & 2j+1 \end{array} \\ \begin{bmatrix} c_j & & s_j \\ & 1 & \\ s_j & & -c_j \end{bmatrix} \end{array} \begin{bmatrix} \bar{\zeta}_j \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \zeta_j \\ \bar{\zeta}_{j+1} \\ \omega_j \end{bmatrix},$$

where $\zeta_j := \bar{c}_j c_j \bar{\zeta}_j$, $\bar{\zeta}_{j+1} := \bar{s}_j c_j \bar{\zeta}_j$, $\omega_j := s_j \bar{\zeta}_j$, and where we initialize $\bar{\zeta}_1 := \beta_1$. The value $\bar{\zeta}_{j+1}$ will be replaced by ζ_{j+1} by the next Givens rotation.

A consequence of the rotation above is that the subproblem (6.2) may equivalently be rewritten

$$\underset{\bar{\mathbf{y}} \in \mathbb{R}^k}{\text{minimize}} \ \tfrac{1}{2} \left\| \begin{bmatrix} \mathbf{R}_k \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \bar{\mathbf{y}} - \begin{bmatrix} \mathbf{z}_k \\ \bar{\zeta}_{k+1} \\ \mathbf{w}_k \end{bmatrix} \right\|_2^2,$$

whose solution is readily identified, $\bar{\mathbf{y}}_k := \mathbf{R}_k^{-1} \mathbf{z}_k$, and whose residual is the norm of $(\bar{\zeta}_{k+1}, \mathbf{w}_k)$.

Since $\bar{\mathbf{y}}_k$ is the solution of an upper triangular system, all of its components likely change at each iteration. Fortunately, it possible to update $\mathbf{y}_k$ directly without requiring $\bar{\mathbf{y}}_k$. Following Paige and Saunders (1982), let $\mathbf{d}_j$ be the j-th column of

$$\mathbf{D}_k := \mathbf{V}_k \mathbf{R}_k^{-1}. \tag{6.6}$$

Upon rearranging, we have $\mathbf{R}_k^T \mathbf{D}_k^T = \mathbf{V}_k^T$ so that we find the rows of $\mathbf{D}_k^T$, i.e., the vectors $\mathbf{d}_j$, recursively:

$$\mathbf{d}_1 = \frac{1}{\rho_1} \mathbf{v}_1, \qquad \mathbf{d}_{j+1} = \frac{1}{\rho_{j+1}} (\mathbf{v}_{j+1} - \theta_{j+1} \mathbf{d}_j), \quad (j = 1, \ldots, k-1). \tag{6.7}$$

Consequently,

$$\mathbf{y}_k = \mathbf{V}_k \bar{\mathbf{y}}_k = \mathbf{V}_k \mathbf{R}_k^{-1} \mathbf{z}_k = \mathbf{D}_k \mathbf{z}_k = \mathbf{y}_{k-1} + \zeta_k \mathbf{d}_k. \tag{6.8}$$

The following result shows that $\mathbf{D}_k$ is a partial factor of $\mathbf{A}^T \mathbf{M}^{-1} \mathbf{A} + \mathbf{N}$ and that the latter matrix defines the appropriate norm to measure direct errors.

Theorem 6.2. *Let the vectors* $\mathbf{d}_k$ *be updated according to (6.7). Then, for* $k = 1, \ldots, m$, *we have*

$$\mathbf{D}_k^{\mathsf{T}}(\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A} + \mathbf{N})\mathbf{D}_k = \mathbf{I}_k. \tag{6.9}$$

In particular,

$$\mathbf{y}_k = \sum_{j=1}^{k} \zeta_j \mathbf{d}_j, \tag{6.10a}$$

$$\|\mathbf{y}\|^2_{\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A}+\mathbf{N}} = \sum_{j=1}^{m} \zeta_j^2 \tag{6.10b}$$

$$\|\mathbf{y} - \mathbf{y}_k\|^2_{\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A}+\mathbf{N}} = \sum_{j=k+1}^{m} \zeta_j^2, \tag{6.10c}$$

where $\mathbf{y}$ *is the solution of (2.4).*

Proof. We have from (4.1), (6.4), and (6.6) that

$$\begin{aligned}
\mathbf{D}_k^{\mathsf{T}}(\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A} + \mathbf{N})\mathbf{D}_k &= \mathbf{R}_k^{-\mathsf{T}}\mathbf{V}_k^{\mathsf{T}}(\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A} + \mathbf{N})\mathbf{V}_k\mathbf{R}_k^{-1} \\
&= \mathbf{R}_k^{-\mathsf{T}}(\mathbf{V}_k^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A}\mathbf{V}_k + \mathbf{I}_k)\mathbf{R}_k^{-1} \\
&= \mathbf{R}_k^{-\mathsf{T}}(\mathbf{E}_k^{\mathsf{T}}\mathbf{U}_{k+1}^{\mathsf{T}}\mathbf{M}\mathbf{U}_{k+1}\mathbf{E}_k + \mathbf{I}_k)\mathbf{R}_k^{-1} \\
&= \mathbf{R}_k^{-\mathsf{T}}\begin{bmatrix}\mathbf{E}_k^{\mathsf{T}} & \mathbf{I}_k\end{bmatrix}\begin{bmatrix}\mathbf{E}_k \\ \mathbf{I}_k\end{bmatrix}\mathbf{R}_k^{-1} \\
&= \mathbf{Q}_k^{\mathsf{T}}\mathbf{Q}_k = \mathbf{I}_k,
\end{aligned}$$

which establishes (6.9). Formulae (6.10a), (6.10b), and (6.10c) follow easily from (6.9) and (6.8). □

Truncating the sum in (6.10b) and (6.10c) yields lower bounds on $\|\mathbf{y}\|^2_{\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A}+\mathbf{N}}$ and $\|\mathbf{y} - \mathbf{y}_k\|^2_{\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A}+\mathbf{N}}$. In particular, given an integer d sufficiently large and an iteration $k \geq d$, the sum over the most recent d iterations

$$\sum_{j=k-d+1}^{k} \zeta_j^2 \leq \|\mathbf{y} - \mathbf{y}_{k-d+1}\|^2_{\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A}+\mathbf{N}} \tag{6.11}$$

can be a good approximation of the direct error between the exact solution $\mathbf{y}$ and the iterate $\mathbf{y}_{k-d+1}$ in the energy norm defined by $\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A}+\mathbf{N}$. Analogously to the approach used by Arioli (2010), Hestenes and Stiefel (1952), Golub and Meurant (1997), Golub and Meurant (2010) and others, this suggests a stopping criterion where we stop the iterations when the partial sum above falls below a tolerance τ^2 times $\sum_{j=1}^{k} \zeta_j^2$.

The update of $\mathbf{d}_{k+1}$ appears to require knowledge of ρ_{k+1}, which is not available during the k-th iteration. It is possible to circumvent this by defining $\mathbf{h}_k := \rho_k \mathbf{d}_k$. We then initialize $\mathbf{h}_1 := \mathbf{v}_1$ and update according to $\mathbf{h}_{k+1} = \mathbf{v}_{k+1} - \theta_{k+1}/\rho_k\, \mathbf{h}_k$. The update of $\mathbf{y}_k$ becomes $\mathbf{y}_k = \mathbf{y}_{k-1} + \zeta_k/\rho_k\, \mathbf{h}_k$. The main steps of G-LSQR are summarized as Algorithm 6.1.

Algorithm 6.1 Generalized LSQR

Require: $\mathbf{M}$, $\mathbf{A}$, $\mathbf{N}$, $\mathbf{b}$, d, τ, k_{max}
$\beta_1\mathbf{M}\mathbf{u}_1 = \mathbf{b}, \quad \alpha_1\mathbf{N}\mathbf{v}_1 = \mathbf{A}^\mathsf{T}\mathbf{u}_1$ *// Initialize bidiagonalization*
$\mathbf{h}_1 = \mathbf{v}_1, \quad \bar{\zeta}_1 = \beta_1, \quad \hat{\alpha}_1 = \alpha_1, \quad \mathbf{y}_0 = \mathbf{0}$
$k = 1, \quad \Delta = 0, \quad$ converged $=$ **false**
while not converged **and** $k < k_{max}$ **do**
  // Continue bidiagonalization
  $\beta_{k+1}\mathbf{M}\mathbf{u}_{k+1} = \mathbf{A}\mathbf{v}_k - \alpha_k\mathbf{M}\mathbf{u}_k, \quad \alpha_{k+1}\mathbf{N}\mathbf{v}_{k+1} = \mathbf{A}^\mathsf{T}\mathbf{u}_{k+1} - \beta_{k+1}\mathbf{N}\mathbf{v}_k$
  $\hat{\rho}_k = (1 + \bar{\rho}_k^2)^{\frac{1}{2}}, \quad c_k = \hat{\alpha}_k/\hat{\rho}_k, \quad s_k = 1/\hat{\rho}_k$ *// Rotation of type II*
  $\rho_k = (\hat{\rho}_k^2 + \beta_{k+1}^2)^{\frac{1}{2}}, \quad \bar{c}_k = \hat{\rho}_k/\rho_k, \quad \bar{s}_k = \beta_{k+1}/\rho_k$ *// Rotation of type I*
  $\theta_{k+1} = \bar{s}_k\alpha_{k+1}, \quad \hat{\alpha}_{k+1} = -\hat{c}_k\alpha_{k+1}$
  $\zeta_k = \bar{c}_k c_k \bar{\zeta}_k, \quad \bar{\zeta}_{k+1} = \bar{s}_k c_k \bar{\zeta}_k, \quad \omega_k = s_k\bar{\zeta}_k$ *// Update solution and residual*
  $\mathbf{y}_k = \mathbf{y}_{k-1} + \zeta_k/\rho_k\,\mathbf{h}_k$ *// Update*
  $\mathbf{h}_{k+1} = \mathbf{v}_{k+1} - \theta_{k+1}/\rho_k\,\mathbf{h}_k,$
  $\Delta = \Delta + \zeta_k^2$
  if $k \geq d$ **then**
    converged $= \left(\sum_{j=k-d+1}^{k} \zeta_j^2 < \tau^2\Delta\right)$ *// Test convergence*
  $k \leftarrow k + 1$
$\mathbf{y} = \mathbf{y}_k$
$\mathbf{x} = \mathbf{M}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{y})$
return $(\mathbf{x}, \mathbf{y})$

6.3. **Generalized Craig.** The Generalized Craig method seeks a solution to the least-norm problem (2.8), which, we reiterate, it perfectly equivalent to the least-squares problem (2.4). After k steps of Algorithm 4.2, we seek approximations

$$\mathbf{x} \approx \mathbf{x}_k := \mathbf{U}_k\bar{\mathbf{x}}_k, \qquad \text{and} \qquad \mathbf{y} \approx \mathbf{y}_k := \mathbf{V}_k\bar{\mathbf{y}}_k$$

for some $\bar{\mathbf{x}}_k \in \mathbb{R}^k$ and $\bar{\mathbf{y}}_k \in \mathbb{R}^k$. In doing so, we have $\|\mathbf{y}\|_\mathbf{N}^2 \approx \|\mathbf{y}_k\|_\mathbf{N}^2 = \|\bar{\mathbf{y}}_k\|^2$ and $\|\mathbf{x}\|_\mathbf{M}^2 \approx \|\mathbf{x}_k\|_\mathbf{M}^2 = \|\bar{\mathbf{x}}_k\|^2$. Moreover,

$$\begin{aligned}\mathbf{M}\mathbf{x} + \mathbf{A}\mathbf{y} &\approx \mathbf{M}\mathbf{x}_k + \mathbf{A}\mathbf{y}_k \\ &= \mathbf{M}\mathbf{U}_k\bar{\mathbf{x}}_k + \mathbf{A}\mathbf{V}_k\bar{\mathbf{y}}_k \\ &= \mathbf{M}\mathbf{U}_k\bar{\mathbf{x}}_k + \mathbf{M}\mathbf{U}_k\mathbf{B}_k\bar{\mathbf{y}}_k + \beta_{k+1}\mathbf{M}\mathbf{u}_{k+1}\mathbf{e}_k^\mathsf{T}\bar{\mathbf{y}}_k.\end{aligned}$$

Upon premultiplying by $\mathbf{U}_k^\mathsf{T}$, the right-hand side becomes $\bar{\mathbf{x}}_k + \mathbf{B}_k\bar{\mathbf{y}}_k$. Therefore, $(\bar{\mathbf{x}}_k, \bar{\mathbf{y}}_k)$ solves

$$\underset{\bar{\mathbf{x}},\bar{\mathbf{y}}}{\text{minimize}} \ \tfrac{1}{2}(\|\bar{\mathbf{x}}\|^2 + \|\bar{\mathbf{y}}\|^2) \quad \text{subject to } \bar{\mathbf{x}} + \mathbf{B}_k\bar{\mathbf{y}} = \beta_1\mathbf{e}_1, \tag{6.12}$$

where we again used the fact that $\mathbf{b} = \mathbf{M}\mathbf{U}_k(\beta_1\mathbf{e}_1)$ for all k. Since the constraints of the latter problem always form a compatible system, Craig's method may be applied. As in the previous section, the k-th subproblem solved by this generalized Craig method is identical to that solved by the *Extended* Craig *Algorithm* of Saunders (1995) with the regularization parameter set to one. The only part of the implementation that need be modified is the bidiagonalization step.

By contrast with §6.1, the solution of (6.12) solves the SQD subsystem

$$\begin{bmatrix} \mathbf{I}_k & \mathbf{B}_k \\ \mathbf{B}_k^{\mathsf{T}} & -\mathbf{I}_k \end{bmatrix} \begin{bmatrix} \bar{\mathbf{x}}_k \\ \bar{\mathbf{y}}_k \end{bmatrix} = \begin{bmatrix} \beta_1 \mathbf{e}_1 \\ \mathbf{0} \end{bmatrix} \tag{6.13}$$

which represents the optimality conditions of the regularized least-squares problem

$$\underset{\bar{\mathbf{y}} \in \mathbb{R}^k}{\text{minimize}} \ \tfrac{1}{2} \left\| \begin{bmatrix} \mathbf{B}_k^{\mathsf{T}} \\ \mathbf{I}_k \end{bmatrix} \bar{\mathbf{y}} - \begin{bmatrix} \beta_1 \mathbf{e}_1 \\ \mathbf{0} \end{bmatrix} \right\|_2^2 .$$

The similarity between the latter and the least-squares problem solved at iteration k of LSQR makes it possible to transfer from the CRAIG point to the LSQR point (Saunders, 1995).

Following Saunders (1995) and Paige (1974), (6.12) may be solved via the LQ factorization of the k-by-$2k$ matrix $\begin{bmatrix} \mathbf{B}_k & \mathbf{I}_k \end{bmatrix}$ by applying $2k-1$ Givens rotations that zero out the identity block. The construction of the rotations is explained below. Their effect is to produce an orthogonal $2k$-by-$2k$ matrix $\mathbf{Q}_k$ and a k-by-k lower bidiagonal matrix $\hat{\mathbf{B}}_k$ such that

$$\begin{bmatrix} \mathbf{B}_k & \mathbf{I}_k \end{bmatrix} \mathbf{Q}_k^{\mathsf{T}} = \begin{bmatrix} \hat{\mathbf{B}}_k & \mathbf{0} \end{bmatrix} . \tag{6.14}$$

We denote

$$\hat{\mathbf{B}}_k := \begin{bmatrix} \hat{\alpha}_1 & & & \\ \hat{\beta}_2 & \hat{\alpha}_2 & & \\ & \ddots & \ddots & \\ & & \hat{\beta}_k & \hat{\alpha}_k \end{bmatrix} . \tag{6.15}$$

Suppose $\mathbf{y}^*$ is the $\mathbf{y}$-component of the exact solution to (1.1) with $\mathbf{f} = \mathbf{b}$ and $\mathbf{g} = \mathbf{0}$ and write $\mathbf{y}^* = \mathbf{V}_m \bar{\mathbf{y}}$ for some vector $\bar{\mathbf{y}}$. The direct error is measured by

$$\|\bar{\mathbf{y}} - \bar{\mathbf{y}}_k\|_2 = \|\mathbf{N}^{\frac{1}{2}} \mathbf{V}_m (\bar{\mathbf{y}} - \mathbf{B}_k^{\mathsf{T}} \bar{\mathbf{z}}_k)\|_2 = \|\mathbf{V}_m (\bar{\mathbf{y}} - \mathbf{B}_k^{\mathsf{T}} \bar{\mathbf{z}}_k)\|_{\mathbf{N}} = \|\mathbf{y}^* - \mathbf{y}_k\|_{\mathbf{N}},$$

where we used the facts that $\bar{\mathbf{y}}_k = \mathbf{B}_k^{\mathsf{T}} \bar{\mathbf{z}}_k$, $\mathbf{y}_k = \mathbf{V}_m \bar{\mathbf{y}}_k$ and the orthogonality of $\mathbf{N}^{\frac{1}{2}} \mathbf{V}_m$. Similarly, $\|\mathbf{x}^* - \mathbf{x}_k\|_{\mathbf{M}} = \|\bar{\mathbf{x}} - \bar{\mathbf{x}}_k\|_2$, where $\mathbf{x}^* = \mathbf{U}_m \bar{\mathbf{x}}$ is the $\mathbf{x}$-component of the exact solution to (1.1). The generalized CRAIG method thus generates a sequence $\{(\mathbf{x}_k, \mathbf{y}_k)\}$ such that $\|\mathbf{x}^* - \mathbf{x}_k\|_{\mathbf{M}}^2 + \|\mathbf{y}^* - \mathbf{y}_k\|_{\mathbf{N}}^2$ is non-increasing.

By definition of the CRAIG method—(Craig, 1955) and (Saunders, 1995, Result 9)—and the observation that

$$\mathbf{M} + \mathbf{A}\mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}} = \mathbf{M}^{\frac{1}{2}} \left(\mathbf{I}_n + (\mathbf{M}^{-\frac{1}{2}} \mathbf{A} \mathbf{N}^{-\frac{1}{2}})(\mathbf{M}^{-\frac{1}{2}} \mathbf{A} \mathbf{N}^{-\frac{1}{2}})^{\mathsf{T}} \right) \mathbf{M}^{\frac{1}{2}},$$

we have the following result.

Theorem 6.3. *The generalized* CRAIG *iterates* $\mathbf{y}_k$ *are related to the iterates* $\mathbf{x}_k$ *of the conjugate gradient method applied to*

$$(\mathbf{M} + \mathbf{A}\mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}})\mathbf{x} = \mathbf{b} \tag{6.16}$$

with preconditioner $\mathbf{M}$ *according to* $\mathbf{y}_k = \mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}}\mathbf{x}_k$.

Proof. Upon multiplying the second block equation of (6.13) by $\mathbf{B}_k$ and substituting the value of $\mathbf{B}_k \bar{\mathbf{y}}_k$ from the first block equation, we obtain

$$(\mathbf{B}_k \mathbf{B}_k^{\mathsf{T}} + \mathbf{I}_k)\bar{\mathbf{x}}_k = \beta_1 \mathbf{e}_1. \tag{6.17}$$

Our subsitution combined with (4.1) yield

$$\mathbf{y}_k = \mathbf{V}_k\bar{\mathbf{y}}_k = \mathbf{V}_k\mathbf{B}_k^{\mathsf{T}}\bar{\mathbf{x}}_k = \mathbf{N}^{-1}\mathbf{B}^{\mathsf{T}}\mathbf{U}_k\bar{\mathbf{x}}_k = \mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}}\mathbf{x}_k.$$

Using the approximation $\mathbf{x} \approx \mathbf{x}_k := \mathbf{U}_k\bar{\mathbf{x}}_k$ in (6.16) and premultiplying with $\mathbf{U}_k^{\mathsf{T}}$, and using the $\mathbf{M}$-orthogonality of the $\mathbf{u}_k$, the $\mathbf{N}$-orthogonality of the $\mathbf{v}_k$ and (4.1), we obtain precisely (6.17).

The system (6.17) may be written equivalently

$$\begin{bmatrix}\mathbf{B}_k & \mathbf{I}_k\end{bmatrix}\begin{bmatrix}\mathbf{B}_k^{\mathsf{T}} \\ \mathbf{I}_k\end{bmatrix}\bar{\mathbf{x}}_k = \beta_1\mathbf{e}_1.$$

Substituting the factorization (6.14) and using the orthogonality of $\mathbf{Q}_k$, we obtain

$$\hat{\mathbf{B}}_k\hat{\mathbf{B}}_k^{\mathsf{T}}\bar{\mathbf{x}}_k = \beta_1\mathbf{e}_1. \tag{6.18}$$

In the latter system, the matrix $\hat{\mathbf{T}}_k := \hat{\mathbf{B}}_k\hat{\mathbf{B}}_k^{\mathsf{T}}$ is tridiagonal, symmetric and positive definite, and its Cholesky factor is $\hat{\mathbf{B}}_k$. It is the tridiagonal matrix generated by a Lanczos process applied to (6.16). Indeed, consider (6.16) in which we substitute $\mathbf{x}$ by an approximation of the form $\mathbf{x}_k = \mathbf{U}_k\bar{\mathbf{x}}_k$. Using (4.1), we have

$$\begin{aligned}(\mathbf{M} + \mathbf{A}\mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}})\mathbf{U}_k &= \mathbf{M}\left(\mathbf{U}_k(\mathbf{I}_k + \mathbf{B}_k\mathbf{B}_k^{\mathsf{T}}) + \beta_{k+1}\mathbf{u}_{k+1}\mathbf{e}_k^{\mathsf{T}}\mathbf{B}_k^{\mathsf{T}}\right) \\ &= \mathbf{M}\left(\mathbf{U}_k\begin{bmatrix}\mathbf{B}_k & \mathbf{I}_k\end{bmatrix}\begin{bmatrix}\mathbf{B}_k^{\mathsf{T}} \\ \mathbf{I}_k\end{bmatrix} + \alpha_k\beta_{k+1}\mathbf{u}_{k+1}\mathbf{e}_k^{\mathsf{T}}\right).\end{aligned} \tag{6.19}$$

It is easy to verify that the k-th off-diagonal element of $\hat{\mathbf{T}}_k = \mathbf{B}_k\mathbf{B}_k^{\mathsf{T}} + \mathbf{I}_k$ is precisely equal to $\alpha_k\beta_{k+1}$. Comparing with (3.2), we conclude that (6.19) corresponds to a Lanczos process on the coefficient matrix of (6.16) with a metric defined by the matrix $\mathbf{M}$. This and the facts established above confirm that the generalized Craig method is equivalent to the conjugate gradient method applied to (6.16) with preconditioner $\mathbf{M}$. □

The proof of Theorem 6.3, and (6.18) in particular, suggests a numerical procedure. Indeed, letting

$$\bar{\mathbf{z}}_k := \hat{\mathbf{B}}_k^{\mathsf{T}}\bar{\mathbf{x}}_k, \tag{6.20}$$

solving for the components of $\bar{\mathbf{z}}_k = (\zeta_1, \ldots, \zeta_k)$ is easy:

$$\zeta_1 := \beta_1/\hat{\alpha}_1, \quad \zeta_{i+1} := -\hat{\beta}_{i+1}\zeta_i/\hat{\alpha}_i, \quad (i = 1, \ldots, k-1). \tag{6.21}$$

Solving for $\mathbf{x}_k$ directly, and bypassing $\bar{\mathbf{x}}_k$, may now be done as in §6.2. By definition,

$$\mathbf{x}_k = \mathbf{U}_k\bar{\mathbf{x}}_k = \mathbf{U}_k\hat{\mathbf{B}}_k^{-T}\bar{\mathbf{z}}_k.$$

Since $\hat{\mathbf{B}}_k^{-T}$ is upper bidiagonal, all components of $\hat{\mathbf{B}}_k^{-T}\bar{\mathbf{z}}_k$ are likely to change at every iteration. Fortunately, upon defining $\mathbf{D}_k := \mathbf{U}_k\hat{\mathbf{B}}_k^{-T}$, and denoting $\mathbf{d}_i$ the i-th column of $\mathbf{D}_k$, we are able to use a recursion formula for $\mathbf{x}_k$ provided that $\mathbf{d}_i$ may be found easily. Slightly rearranging, we have

$$\hat{\mathbf{B}}_k\mathbf{D}_k^{\mathsf{T}} = \mathbf{U}_k^{\mathsf{T}}$$

and therefore it is easy to identify each $\mathbf{d}_i$—i.e., each row of $\mathbf{D}_k^{\mathsf{T}}$—recursively:

$$\mathbf{d}_1 := \mathbf{u}_1/\hat{\alpha}_1, \quad \mathbf{d}_{i+1} := (\mathbf{u}_{i+1} - \hat{\beta}_{i+1}\mathbf{d}_i)/\hat{\alpha}_{i+1}, \quad (i = 1, \ldots, k-1). \tag{6.22}$$

This yields the update

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \zeta_{k+1}\mathbf{d}_{k+1} \tag{6.23}$$

for $\mathbf{x}_{k+1}$. In the next section, we return to the expansion of each $\mathbf{x}_k$ as a linear combination of the columns of $\mathbf{D}_k$.

6.4. **Generalized Craig Recursive Expressions.** Saunders (1995) describes an implementation of the extended Craig method in the variable $\mathbf{y}$. In this section, we describe an implementation in the variable $\mathbf{x}$ that resembles that of Arioli (2010).

The $(1, k+1)$-st element of $\begin{bmatrix}\mathbf{B}_k & \mathbf{I}_k\end{bmatrix}$ (equal to 1), may be zeroed out by applying a Givens rotation acting on columns 1 and $k+1$. By convention, we denote this rotation $\mathbf{Q}_{1,k+1}$, the last index indicating the column of the element being zeroed out, and label it a rotation of type I. As the $(2,1)$-th element of the constraint matrix is $\beta_2 > 0$, the rotation creates a new nonzero element in position $(2, k+1)$, which we denote γ_2. The newly created γ_2 may be zeroed out by a second rotation, of type II, acting on columns $k+1$ and $k+2$, i.e., $\mathbf{Q}_{k+2,k+1}$. Aside from zeroing out γ_2, the effect of $\mathbf{Q}_{k+2,k+1}$ is to change the value 1 in position $(k+2, k+2)$ to some other value, denoted δ_2, which can be zeroed out by a new rotation of type I. With the convention that $\delta_1 = 1$, a general rotation of type I, with the purpose of zeroing out a δ_k, may be represented schematically as (ignoring all irrelevant rows and columns):

$$\begin{array}{c} \\ k \\ k+1 \end{array}\!\!\begin{array}{c} \begin{array}{cc} k & 2k \end{array} \\ \begin{bmatrix} \alpha_k & \delta_k \\ \beta_{k+1} & 0 \end{bmatrix} \end{array}\!\!\begin{array}{c} \begin{array}{cc} k & 2k \end{array} \\ \begin{bmatrix} c_k & s_k \\ s_k & -c_k \end{bmatrix} \end{array} = \begin{bmatrix} \rho_k & 0 \\ c_k\beta_{k+1} & \gamma_{k+1} \end{bmatrix} := \begin{bmatrix} \hat{\alpha}_k & 0 \\ \hat{\beta}_{k+1} & \gamma_{k+1} \end{bmatrix},$$

where $\rho_k := \sqrt{\alpha_k^2 + \delta_k^2}$, $c_k := \alpha_k/\rho_k$, $s_k := \delta_k/\rho_k$, $\hat{\beta}_{k+1} := c_k\beta_{k+1}$, and $\gamma_{k+1} := s_k\beta_{k+1}$, and where labels to the left and above a matrix indicate row and column indices, respectively. It is easy to show by induction that $\hat{\alpha}_k > 0$ and therefore that the procedure (6.21) is well defined. Similarly, a general rotation of type II, with the purpose of zeroing out a γ_{k+1}, may be represented schematically as

$$\begin{array}{c} \\ k+1 \\ k+2 \end{array}\!\!\begin{array}{c} \begin{array}{cc} 2k+1 & 2k+2 \end{array} \\ \begin{bmatrix} \gamma_{k+1} & 1 \\ 0 & 0 \end{bmatrix} \end{array}\!\!\begin{array}{c} \begin{array}{cc} 2k+1 & 2k+2 \end{array} \\ \begin{bmatrix} \bar{c}_k & \bar{s}_k \\ \bar{s}_k & -\bar{c}_k \end{bmatrix} \end{array} = \begin{bmatrix} 0 & \delta_{k+1} \\ 0 & 0 \end{bmatrix},$$

where $\bar{c}_k := -1/\sqrt{\gamma_{k+1}^2 + 1}$, $\bar{s}_k := \gamma_{k+1}/\sqrt{\gamma_{k+1}^2 + 1}$, and $\delta_{k+1} := \bar{s}_k\gamma_{k+1} - \bar{c}_k = \sqrt{\gamma_{k+1}^2 + 1}$. It is now not too difficult to see that the sequence of rotations required to perform the LQ factorization is given by

$$\mathbf{Q}_k^T := (\mathbf{Q}_{1,k+1}\mathbf{Q}_{k+2,k+1})(\mathbf{Q}_{2,k+2}\mathbf{Q}_{k+3,k+2})\cdots(\mathbf{Q}_{k-1,2k-1}\mathbf{Q}_{2k,2k-1})\mathbf{Q}_{k,2k}.$$

Other rotations are possible; the one given coincides with that of Saunders (1995).

We can verify by induction that

$$\begin{aligned} \hat{\alpha}_1 &= \sqrt{\alpha_1^2 + 1}, \\ \hat{\alpha}_{k+1} &= \sqrt{\alpha_{k+1}^2 + \delta_{k+1}^2}, \quad k \geq 2, \\ \hat{\beta}_{k+1} &= \alpha_k\beta_{k+1}/\hat{\alpha}_k, \quad k \geq 2. \end{aligned} \tag{6.24}$$

At this point, we have constructed $\hat{\mathbf{B}}_k$ and we may update the vectors $\mathbf{d}_k$ and the estimate $\mathbf{x}_k$ as in (6.22) and (6.23).

The factorization (6.14) implicitly fixes $\bar{\mathbf{y}}_k = \mathbf{B}_k^T\bar{\mathbf{x}}_k$ in (6.13).

At each iteration, the residual of (1.1) at $(\mathbf{x}_k, \mathbf{y}_k)$,

$$\begin{bmatrix} \mathbf{r}_k \\ \mathbf{s}_k \end{bmatrix} := \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & -\mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{y}_k \end{bmatrix} \tag{6.25}$$

is given by

$$\begin{aligned} \begin{bmatrix} \mathbf{r}_k \\ \mathbf{s}_k \end{bmatrix} &= \begin{bmatrix} \beta_1 \mathbf{M}\mathbf{u}_1 \\ 0 \end{bmatrix} - \begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & -\mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{U}_k & \\ & \mathbf{V}_k \end{bmatrix} \begin{bmatrix} \bar{\mathbf{x}}_k \\ \bar{\mathbf{y}}_k \end{bmatrix} \\ &= \begin{bmatrix} \beta_1 \mathbf{M}\mathbf{u}_1 \\ 0 \end{bmatrix} - \begin{bmatrix} \mathbf{M} & \\ & \mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{U}_k & \\ & \mathbf{V}_k \end{bmatrix} \begin{bmatrix} \mathbf{I}_k & \mathbf{B}_k \\ \mathbf{B}_k^T & -\mathbf{I}_k \end{bmatrix} \begin{bmatrix} \bar{\mathbf{x}}_k \\ \bar{\mathbf{y}}_k \end{bmatrix} - \begin{bmatrix} \beta_{k+1} \mathbf{M}\mathbf{u}_{k+1} \\ \mathbf{0} \end{bmatrix} \mathbf{e}_{2k}^T \begin{bmatrix} \bar{\mathbf{x}}_k \\ \bar{\mathbf{y}}_k \end{bmatrix} \\ &= \begin{bmatrix} -\beta_{k+1}\eta_k \mathbf{M}\mathbf{u}_{k+1} \\ \mathbf{0} \end{bmatrix}, \end{aligned}$$

where we used (4.1), (6.13) and where η_k is the k-th component of $\bar{\mathbf{y}}_k$. The vector $\bar{\mathbf{y}}_k$ is not directly available but η_k can nonetheless be monitored cheaply since, by definition,

$$\eta_k = \mathbf{e}_k^T \begin{bmatrix} \bar{\mathbf{y}}_k \\ \bar{\mathbf{x}}_k \end{bmatrix} = \mathbf{e}_k^T \mathbf{Q}_k^T \begin{bmatrix} \bar{\mathbf{z}}_k \\ \mathbf{0} \end{bmatrix} = \mathbf{e}_k^T \mathbf{Q}_{k,2k} \begin{bmatrix} \bar{\mathbf{z}}_k \\ \mathbf{0} \end{bmatrix} = c_k \zeta_k, \tag{6.26}$$

where ζ_k is the k-th component of $\bar{\mathbf{z}}_k$, while c_k is the cosine defining $\mathbf{Q}_{k,2k}$. This establishes that

$$\|\mathbf{r}_k\|_{\mathbf{M}^{-1}} = |\beta_{k+1} c_k \zeta_k|. \tag{6.27}$$

The residual of (6.16) may be monitored similarly. As in (6.19), we have

$$\begin{aligned} \mathbf{q}_k &:= \mathbf{b} - (\mathbf{M} + \mathbf{A}\mathbf{N}^{-1}\mathbf{A}^T)\mathbf{U}_k \bar{\mathbf{x}}_k \\ &= \beta_1 \mathbf{M}\mathbf{u}_1 - \mathbf{M}\left(\mathbf{U}_k(\mathbf{I}_k + \mathbf{B}_k\mathbf{B}_k^T)\bar{\mathbf{x}}_k + \alpha_k \beta_{k+1} \mathbf{u}_{k+1} \mathbf{e}_k^T \bar{\mathbf{x}}_k\right) \\ &= \alpha_k \beta_{k+1} \xi_k \mathbf{M}\mathbf{u}_{k+1}. \end{aligned} \tag{6.28}$$

where we used (6.17) and we denoted ξ_k the last component of $\bar{\mathbf{x}}_k$. As before, $\bar{\mathbf{x}}_k$ is not directly accessible but, by definition,

$$\xi_k = \mathbf{e}_{2k}^T \begin{bmatrix} \bar{\mathbf{y}}_k \\ \bar{\mathbf{x}}_k \end{bmatrix} = \mathbf{e}_{2k}^T \mathbf{Q}_k^T \begin{bmatrix} \bar{\mathbf{z}}_k \\ \mathbf{0} \end{bmatrix} = \mathbf{e}_{2k}^T \mathbf{Q}_{2k,2k-1} \mathbf{Q}_{k,2k} \begin{bmatrix} \bar{\mathbf{z}}_k \\ \mathbf{0} \end{bmatrix} = s_k \zeta_k, \tag{6.29}$$

where we have used the facts that $\mathbf{e}_{2k}^T \mathbf{Q}_{2k,2k-1} = \bar{s}_k \mathbf{e}_{2k-1}^T - \bar{c}_k \mathbf{e}_{2k}^T$, that $\mathbf{e}_{2k-1}^T \mathbf{Q}_{k,2k} = \mathbf{0}$, and $\mathbf{e}_{2k}^T \mathbf{Q}_{k,2k} = s_k \mathbf{e}_k^T - c_k \mathbf{e}_{2k}^T$. Since $\mathbf{q}_k \in \mathbb{M}^\star$, we have, using the $\mathbf{M}$-orthogonality of $\mathbf{u}_{k+1}$,

$$\|\mathbf{q}_k\|_{\mathbf{M}^{-1}} = |\alpha_k \beta_{k+1} s_k \zeta_k|. \tag{6.30}$$

Note that from (6.21), (6.26) and (6.29), we also have

$$\xi_k^2 + \eta_k^2 = \zeta_k^2 \quad \text{for all } k. \tag{6.31}$$

The generalized Craig method is summarized as Algorithm 6.2.

As in the case of Lsqr, Algorithm 6.2 implicitly constructs and updates a partial factorization of a matrix determining the energy norm.

Theorem 6.4. *Let* $\hat{\mathbf{B}}_k$ *be defined by (6.14) and* $\mathbf{D}_k := \mathbf{U}_k \hat{\mathbf{B}}_k^{-T}$. *For* $k = 1, \ldots, n$, *we have*

$$\mathbf{D}_k^{\mathsf{T}}(\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}} + \mathbf{M})\mathbf{D}_k = \mathbf{I}_k. \tag{6.32}$$

In particular,

$$\mathbf{x}_k = \sum_{j=1}^{k} \zeta_j \mathbf{d}_j \tag{6.33}$$

and we have the estimates

$$\|\mathbf{x}_k\|^2_{\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}}+\mathbf{M}} = \sum_{i=1}^{k} \zeta_i^2, \tag{6.34a}$$

$$\|\mathbf{x}^* - \mathbf{x}_k\|^2_{\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}}+\mathbf{M}} = \sum_{i=k+1}^{n} \zeta_i^2, \tag{6.34b}$$

$$\|\mathbf{x}_k\|^2_{\mathbf{M}} = \sum_{i=1}^{k} \xi_i^2, \tag{6.34c}$$

$$\|\mathbf{x}^* - \mathbf{x}_k\|^2_{\mathbf{M}} = \sum_{i=k+1}^{n} \xi_i^2, \tag{6.34d}$$

where ζ_i *and* ξ_i *are as in (6.21) and (6.29).*
In addition, the residuals $\mathbf{r}_k$ *and* $\mathbf{q}_k$ *defined in* (6.25) *and* (6.28) *satisfy*

$$\|\mathbf{r}_k\|_{\mathbf{M}^{-1}} \le \|\bar{\mathbf{A}}\|_2 \, |\zeta_k| \quad \textit{and} \quad \|\mathbf{q}_k\|_{\mathbf{M}^{-1}} \le \|\bar{\mathbf{A}}\|_2^2 \, |\zeta_k|. \tag{6.35}$$

Proof. We have from (6.14), the orthogonality of $\mathbf{Q}_k$, the $\mathbf{M}$-orthogonality of the $\mathbf{u}_k$, the $\mathbf{N}$-orthogonality of the $\mathbf{v}_k$ and (4.1), that

$$\begin{aligned}
\mathbf{D}_k^{\mathsf{T}}(\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}} + \mathbf{M})\mathbf{D}_k &= \hat{\mathbf{B}}_k^{-1}\mathbf{U}_k^{\mathsf{T}}(\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}} + \mathbf{M})\mathbf{U}_k\hat{\mathbf{B}}_k^{-T} \\
&= \hat{\mathbf{B}}_k^{-1}(\mathbf{U}_k^{\mathsf{T}}\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}}\mathbf{U}_k + \mathbf{I}_k)\hat{\mathbf{B}}_k^{-T} \\
&= \hat{\mathbf{B}}_k^{-1}(\mathbf{B}_k\mathbf{V}_k^{\mathsf{T}}\mathbf{N}\mathbf{V}_k\mathbf{B}_k^{\mathsf{T}} + \mathbf{I}_k)\hat{\mathbf{B}}_k^{-T} \\
&= \hat{\mathbf{B}}_k^{-1}(\mathbf{B}_k\mathbf{B}_k^{\mathsf{T}} + \mathbf{I}_k)\hat{\mathbf{B}}_k^{-T} \\
&= \hat{\mathbf{B}}_k^{-1}\begin{bmatrix}\mathbf{B}_k & \mathbf{I}_k\end{bmatrix}\begin{bmatrix}\mathbf{B}_k^{\mathsf{T}} \\ \mathbf{I}_k\end{bmatrix}\hat{\mathbf{B}}_k^{-T} \\
&= \hat{\mathbf{B}}_k^{-1}\begin{bmatrix}\hat{\mathbf{B}}_k & \mathbf{0}\end{bmatrix}\mathbf{Q}_k\mathbf{Q}_k^{\mathsf{T}}\begin{bmatrix}\hat{\mathbf{B}}_k^{\mathsf{T}} \\ \mathbf{0}\end{bmatrix}\hat{\mathbf{B}}_k^{-T} \\
&= \hat{\mathbf{B}}_k^{-1}\hat{\mathbf{B}}_k\hat{\mathbf{B}}_k^{\mathsf{T}}\hat{\mathbf{B}}_k^{-T} \\
&= \mathbf{I}_k,
\end{aligned}$$

which establishes (6.32).

Using the update formula (6.23) for $\mathbf{x}_k$, we may write $\mathbf{x}_k = \mathbf{D}_k\bar{\mathbf{z}}_k$. Thus,

$$\|\mathbf{x}_k\|^2_{\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}}+\mathbf{M}} = \mathbf{x}_k^{\mathsf{T}}(\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}} + \mathbf{M})\mathbf{x}_k = \bar{\mathbf{z}}_k^{\mathsf{T}}\mathbf{D}_k^{\mathsf{T}}(\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}} + \mathbf{M})\mathbf{D}_k\bar{\mathbf{z}}_k = \|\bar{\mathbf{z}}_k\|_2^2.$$

Therefore

$$\|\mathbf{x}_k\|^2_{\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^T+\mathbf{M}} = \sum_{i=1}^{k} \zeta_i^2,$$

and this establishes (6.34a). Since

$$\|\mathbf{x}^*\|^2_{\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^T+\mathbf{M}} = \|\mathbf{x}_n\|^2_{\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^T+\mathbf{M}} = \sum_{i=1}^{n} \zeta_i^2,$$

we obtain (6.34b) as before.

Since $\mathbf{M}\mathbf{x}_k = \mathbf{M}\mathbf{U}_k\bar{\mathbf{x}}_k$ and $\|\bar{\mathbf{x}}_k\|_2^2 = \sum_{i=1}^{k} \xi_i^2$ by definition of ξ_i, we obtain the second equality in (6.34c).

Using

$$\|\mathbf{x}^*\|^2_{\mathbf{M}} = \|\mathbf{x}_n\|^2_{\mathbf{M}} = \sum_{i=1}^{n} \xi_i^2,$$

we have the error estimate

$$\|\mathbf{x}^* - \mathbf{x}_k\|^2_{\mathbf{M}} = \sum_{i=k+1}^{n} \xi_i^2,$$

which is identical to (6.34d).

The proof of (6.35) follows directly form (6.25), (6.28) and (Arioli, 2010, Proposition 3.1). □

As in §6.2, (6.34b) suggests a stopping condition of the form

$$\sum_{i=k+1}^{k+d+1} \zeta_i^2 \leq \tau^2 \sum_{i=1}^{k+d+1} \zeta_i^2,$$

for a user-chosen integer $d \in \mathbb{N}_0$ and $\tau \in (0,1)$.

In Theorem 6.4, the quantity $\|\bar{\mathbf{A}}\|_2$ is the largest singular value of $\bar{\mathbf{A}}$, which coincides with the largest elliptic singular value of $\mathbf{A}$.

6.5. **Generalized Lsmr.** Lsmr (Fong and Saunders, 2011) consists in applying Minres (Paige and Saunders, 1975) to the normal equations (2.6). Since the appropriate norm for measuring the residual of the normal equations is the $\mathbf{N}^{-1}$-norm, we premultiply (2.6) by $\mathbf{N}^{-\frac{1}{2}}$:

$$\mathbf{N}^{-\frac{1}{2}}(\mathbf{A}^T\mathbf{M}^{-1}\mathbf{A} + \mathbf{N})\mathbf{y} = \mathbf{N}^{-\frac{1}{2}}\mathbf{A}^T\mathbf{M}^{-1}\mathbf{b}. \tag{6.36}$$

By definition, Minres then computes $\mathbf{y}$ so as to

$$\underset{\mathbf{y}\in\mathbb{R}^m}{\text{minimize}} \ \tfrac{1}{2}\|\mathbf{N}^{-\frac{1}{2}}(\mathbf{A}^T\mathbf{M}^{-1}\mathbf{b} - (\mathbf{A}^T\mathbf{M}^{-1}\mathbf{A} + \mathbf{N})\mathbf{y}))\|_2.$$

Seeking again an approximation $\mathbf{y} \approx \mathbf{y}_k := \mathbf{V}_k\bar{\mathbf{y}}_k$ and using (4.1), we have

$$\mathbf{M}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{y}) = \mathbf{M}^{-1}(\mathbf{M}\mathbf{U}_{k+1}(\beta_1\mathbf{e}_1) - \mathbf{A}\mathbf{V}_k\bar{\mathbf{y}}_k) = \mathbf{U}_{k+1}(\beta_1\mathbf{e}_1 - \mathbf{E}_k\bar{\mathbf{y}}_k)$$

and

$$\mathbf{A}^T\mathbf{M}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{y}) = (\mathbf{N}\mathbf{V}_k\mathbf{E}_k^T + \alpha_{k+1}\mathbf{N}\mathbf{v}_{k+1}\mathbf{e}_{k+1}^T)(\beta_1\mathbf{e}_1 - \mathbf{E}_k\bar{\mathbf{y}}_k).$$

Using the identities

$$\mathbf{e}_{k+1}^T\mathbf{e}_1 = 0, \quad \mathbf{e}_{k+1}^T\mathbf{E}_k = \beta_{k+1}\mathbf{e}_k^T, \quad \text{and} \quad \mathbf{E}_k^T\mathbf{e}_1 = \alpha_1\mathbf{e}_1,$$

Algorithm 6.2 Generalized CRAIG

Require: $\mathbf{M}$, $\mathbf{A}$, $\mathbf{N}$, $\mathbf{b}$, $\mathbf{d}$, τ, k_{max}
// Initialization
$\beta_1\mathbf{M}\mathbf{u}_1 = \mathbf{b}, \quad \alpha_1\mathbf{N}\mathbf{v}_1 = \mathbf{A}^\mathsf{T}\mathbf{u}_1$ *// Initialize bidiagonalization*
$\delta_1 = 1, \quad \hat{\alpha}_1 = (\alpha_1^2 + 1)^{\frac{1}{2}}, \quad c_1 = \alpha_1/\hat{\alpha}_1, \quad s_1 = 1/\hat{\alpha}_1, \quad \zeta_1 = s_1\beta_1$
$\mathbf{d}_1 = s_1\mathbf{u}_1, \quad \mathbf{x}_1 = \zeta_1\mathbf{d}_1$
$k = 1, \quad \Delta = \zeta_1^2, \quad$ converged = **false**
while not converged **and** $k < k_{max}$ **do**
  // Continue bidiagonalization
  $\beta_{k+1}\mathbf{M}\mathbf{u}_{k+1} = \mathbf{A}\mathbf{v}_k - \alpha_k\mathbf{M}\mathbf{u}_k, \quad \alpha_{k+1}\mathbf{N}\mathbf{v}_{k+1} = \mathbf{A}^\mathsf{T}\mathbf{u}_{k+1} - \beta_{k+1}\mathbf{N}\mathbf{v}_k$
  $\hat{\beta}_{k+1} = c_k\beta_{k+1}, \quad \gamma_{k+1} = s_k\beta_{k+1}$ *// Continue rotation of type I*
  $\delta_{k+1} = (\gamma_{k+1}^2 + 1)^{\frac{1}{2}}, \ \bar{c}_k = -1/\delta_{k+1}, \ \bar{s}_k = \gamma_{k+1}/\delta_{k+1}$ *// Rotation of type II*
  // Compute next Givens rotation of type I
  $\hat{\alpha}_{k+1} = (\alpha_{k+1}^2 + \delta_{k+1}^2)^{\frac{1}{2}}, \quad c_{k+1} = \alpha_{k+1}/\hat{\alpha}_{k+1}, \quad s_{k+1} = \delta_{k+1}/\hat{\alpha}_{k+1}$
  $\zeta_{k+1} = -\hat{\beta}_{k+1}\zeta_k/\hat{\alpha}_{k+1}, \quad \Delta = \Delta + \zeta_{k+1}^2,$ *// Update*
  $\mathbf{d}_{k+1} = (\mathbf{u}_{k+1} - \hat{\beta}_{k+1}\mathbf{d}_k)/\hat{\alpha}_{k+1}$
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \zeta_{k+1}\mathbf{d}_{k+1}$
  if $k \geq d - 1$ **then**
    converged = $\left(\sum_{j=k-d+2}^{k+1} \zeta_j^2 < \tau^2\Delta\right)$ *// Test convergence in* $\mathbf{x}$
  $k \leftarrow k + 1$
$\mathbf{x} = \mathbf{x}_{k+1}$
$\mathbf{y} = \mathbf{N}^{-1}\mathbf{A}^\mathsf{T}\mathbf{x}$
return $(\mathbf{x}, \mathbf{y})$

there remains

$$\mathbf{A}^\mathsf{T}\mathbf{M}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{y}) = \mathbf{N}\mathbf{V}_{k+1}\left(\alpha_1\beta_1\mathbf{e}_1 - \begin{bmatrix}\mathbf{E}_k^\mathsf{T}\mathbf{E}_k \\ \alpha_{k+1}\beta_{k+1}\mathbf{e}_k^\mathsf{T}\end{bmatrix}\bar{\mathbf{y}}_k\right).$$

Similarly,

$$\mathbf{N}\mathbf{y} = \mathbf{N}\mathbf{V}_k\bar{\mathbf{y}}_k = \mathbf{N}\mathbf{V}_{k+1}\begin{bmatrix}\bar{\mathbf{y}}_k \\ \mathbf{0}\end{bmatrix}.$$

By orthogonality of $\mathbf{N}^{\frac{1}{2}}\mathbf{V}_{k+1}$,

$$\|\mathbf{N}^{-\frac{1}{2}}(\mathbf{A}^\mathsf{T}\mathbf{M}^{-1}\mathbf{b} - (\mathbf{A}^\mathsf{T}\mathbf{M}^{-1}\mathbf{A} + \mathbf{N})\mathbf{y}))\|_2 = \left\|\mathbf{N}^{\frac{1}{2}}\mathbf{V}_{k+1}\left(\alpha_1\beta_1\mathbf{e}_1 - \begin{bmatrix}\mathbf{E}_k^\mathsf{T}\mathbf{E}_k + \mathbf{I}_k \\ \alpha_{k+1}\beta_{k+1}\mathbf{e}_k^\mathsf{T}\end{bmatrix}\bar{\mathbf{y}}\right)\right\|_2 = \left\|\alpha_1\beta_1\mathbf{e}_1 - \begin{bmatrix}\mathbf{E}_k^\mathsf{T}\mathbf{E}_k + \mathbf{I}_k \\ \alpha_{k+1}\beta_{k+1}\mathbf{e}_k^\mathsf{T}\end{bmatrix}\bar{\mathbf{y}}\right\|_2. \tag{6.37}$$

Minimizing the latter residual is precisely the subproblem solved by the classic regularized LSMR with parameter $\lambda = 1$. Once again, changing the Golub-Kahan procedure in LSMR is all that is required to solve (1.1) with $\mathbf{f} = \mathbf{b}$ and $\mathbf{g} = \mathbf{0}$.

A by-product of the above is the underlying Lanczos process

$$
\begin{aligned}
(\mathbf{N} + \mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A})\mathbf{V}_{k+1} &= \mathbf{N}\left(\mathbf{V}_k(\mathbf{E}_k^{\mathsf{T}}\mathbf{E}_k + \mathbf{I}_k) + \alpha_{k+1}\beta_{k+1}\mathbf{v}_{k+1}\mathbf{e}_k^{\mathsf{T}}\right), \\
&= \mathbf{N}\mathbf{V}_{k+1}\begin{bmatrix} \mathbf{E}_k^{\mathsf{T}}\mathbf{E}_k + \mathbf{I}_k \\ \alpha_{k+1}\beta_{k+1}\mathbf{e}_k^{\mathsf{T}} \end{bmatrix}
\end{aligned} \tag{6.38}
$$

which we already discovered in the proof of Theorem 6.1 and which is equivalent to what would be generated by applying the standard Lanczos process to $(\mathbf{N} + \mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A})$ with initial vector $\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{b}$ in the metric defined by $\mathbf{N}$. We detail the Lanczos process that characterizes MINRES on this set of normal equations in §8.4.

We have established the following result, which follows directly from the very definition of LSMR and the Lanczos process (6.38).

Theorem 6.5. *The generalized* LSMR *iterates on (2.4) are the same as those generated by the* MINRES *method on the positive definite system (2.6) in the metric defined by* $\mathbf{N}$.

6.6. **Generalized LSMR Recursive Expressions.** Most of the details in this section come directly from (Fong and Saunders, 2011) but will turn out to be useful in designing a stopping condition in the appropriate norm.

The core of G-LSMR revolves around two QR factorizations. The first is the factorization of §6, i.e.,

$$
\mathbf{Q}_k^{\mathsf{T}}\begin{bmatrix} \mathbf{E}_k \\ \mathbf{I}_k \end{bmatrix} = \begin{bmatrix} \mathbf{R}_k \\ \mathbf{0} \end{bmatrix}, \qquad \mathbf{R}_k = \begin{bmatrix} \rho_1 & \theta_2 & & \\ & \rho_2 & \ddots & \\ & & \ddots & \theta_k \\ & & & \rho_k \end{bmatrix}.
$$

As before, $\mathbf{E}_k^{\mathsf{T}}\mathbf{E}_k + \mathbf{I}_k = \mathbf{R}_k^{\mathsf{T}}\mathbf{R}_k$. The least-squares problem (6.37) then reads

$$
\underset{\bar{\mathbf{y}}}{\text{minimize}} \ \tfrac{1}{2}\left\| \alpha_1\beta_1\mathbf{e}_1 - \begin{bmatrix} \mathbf{R}_k^{\mathsf{T}}\mathbf{R}_k \\ \alpha_{k+1}\beta_{k+1}\mathbf{e}_k^{\mathsf{T}} \end{bmatrix}\bar{\mathbf{y}} \right\|_2. \tag{6.39}
$$

Define the approximation $\mathbf{y} \approx \mathbf{y}_k := \mathbf{V}_k\bar{\mathbf{y}}_k$ where $\bar{\mathbf{y}}_k$ solves the above least-squares problem and let $\mathbf{t}_k := \mathbf{R}_k\bar{\mathbf{y}}_k$. Define also $\mathbf{q}_k := \mathbf{R}_k^{-\mathsf{T}}(\alpha_{k+1}\beta_{k+1}\mathbf{e}_k) = \gamma_k\mathbf{e}_k$ where

$$
\gamma_k := \alpha_{k+1}\beta_{k+1}/\rho_k. \tag{6.40}
$$

We now perform a second QR factorization

$$
\bar{\mathbf{Q}}_k\begin{bmatrix} \mathbf{R}_k^{\mathsf{T}} & \alpha_1\beta_1\mathbf{e}_1 \\ \gamma_k\mathbf{e}_k^{\mathsf{T}} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{R}}_k & \mathbf{z}_k \\ \mathbf{0} & \zeta_{k+1} \end{bmatrix}, \qquad \bar{\mathbf{R}}_k = \begin{bmatrix} \bar{\rho}_1 & \bar{\theta}_2 & & \\ & \bar{\rho}_2 & \ddots & \\ & & \ddots & \bar{\theta}_k \\ & & & \bar{\rho}_k \end{bmatrix}.
$$

For this second factorization to be well defined recursively, it is necessary to show that $\gamma_k = \theta_{k+1}$ so that the matrix $\begin{bmatrix} \mathbf{R}_k & \gamma_k\mathbf{e}_k \end{bmatrix}$ is $\mathbf{R}_k$ with one extra column taken from $\mathbf{R}_{k+1}$. But upon examination of the sets of two rotations defining the first

factorization in §6, we see that $\theta_{j+1} = \bar{s}_k \alpha_{j+1} = \alpha_{j+1}\beta_{j+1}/\rho_j = \gamma_j$. The least-squares problem in $\bar{\mathbf{y}}$ may now be restated as

$$\underset{\mathbf{t}}{\text{minimize}} \ \tfrac{1}{2} \left\| \begin{bmatrix} \mathbf{z}_k \\ \zeta_{k+1} \end{bmatrix} - \begin{bmatrix} \bar{\mathbf{R}}_k \\ \mathbf{0} \end{bmatrix} \mathbf{t} \right\|_2 .$$

The solution is obtained by setting $\mathbf{t}_k := \bar{\mathbf{R}}_k^{-1}\mathbf{z}_k$ and the value of the residual is $|\zeta_{k+1}|$.

Since $\bar{\mathbf{R}}_k$ is upper triangular, it is likely that all components of $\mathbf{t}_k$ change from one iteration to the next and it is more efficient to seek an update of $\mathbf{y}_k$ directly. The first step to achieve this is to define the matrix $\mathbf{D}_k$ as in (6.6) and to additionally define

$$\bar{\mathbf{D}}_k := \mathbf{D}_k \bar{\mathbf{R}}_k^{-1}. \tag{6.41}$$

This defines the columns $\bar{\mathbf{d}}_i$ of $\bar{\mathbf{D}}_k$ recursively as the solution of a lower triangular system:

$$\bar{\mathbf{d}}_0 := \mathbf{0}, \quad \text{and} \quad \bar{\mathbf{d}}_{i+1} := \frac{1}{\bar{\rho}_{i+1}}(\mathbf{d}_{i+1} - \bar{\theta}_{i+1}\bar{\mathbf{d}}_i) \quad i \geq 0. \tag{6.42}$$

With these definitions we may update $\mathbf{y}_k$ as follows:

$$\mathbf{y}_k = \mathbf{V}_k \bar{\mathbf{y}}_k = \mathbf{V}_k \mathbf{R}_k^{-1}\mathbf{t}_k = \mathbf{D}_k \mathbf{t}_k = \bar{\mathbf{D}}_k \bar{\mathbf{R}}_k \mathbf{t}_k = \bar{\mathbf{D}}_k \mathbf{z}_k = \mathbf{y}_{k-1} + \zeta_k \bar{\mathbf{d}}_k. \tag{6.43}$$

The matrix $\bar{\mathbf{D}}_k$ yields a partial factorization of the operator that determines the energy norm in which errors should be measured in G-LSMR. The rationale behind this energy norm is the following. Consider temporarily a hypothetical symmetric and positive-definite system $\mathbf{Cx} = \mathbf{b}$. It is the defining property of MINRES that $\|\mathbf{r}_k\|_2$ decreases monotonically, where $\mathbf{r}_k := \mathbf{b} - \mathbf{C}\mathbf{x}_k = \mathbf{C}(\mathbf{x}_* - \mathbf{x}_k)$, and where $\mathbf{x}_*$ is the unique solution of the system. Thus $\|\mathbf{r}_k\|_2 = \|\mathbf{x}_* - \mathbf{x}_k\|_{\mathbf{C}^2}$ is the quantity that decreases. Suppose now, as in (6.36) that the residual must be measured in the $\mathbf{N}^{-1}$-norm. Then

$$\|\mathbf{r}_k\|_{\mathbf{N}^{-1}} = \|\mathbf{N}^{-\frac{1}{2}}\mathbf{r}_k\|_2 = \|\mathbf{N}^{-\frac{1}{2}}\mathbf{C}(\mathbf{x}_* - \mathbf{x}_k)\| = \|\mathbf{x}_* - \mathbf{x}_k\|_{\mathbf{C}\mathbf{N}^{-1}\mathbf{C}}$$

is the appropriate energy norm in which the error should be measured in MINRES. The next result summarizes the main properties in our SQD context.

Theorem 6.6. *Let* $\bar{\mathbf{D}}_k$ *be defined as in* (6.41). *Then, for* $k = 1, \dots, m$, *we have*

$$\bar{\mathbf{D}}_k^{\mathsf{T}} \mathbf{G} \bar{\mathbf{D}}_k = \mathbf{I}_k, \quad \text{where } \mathbf{G} := (\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A} + \mathbf{N})\mathbf{N}^{-1}(\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A} + \mathbf{N}). \tag{6.44}$$

In particular,

$$\mathbf{y}_k = \sum_{j=1}^{k} \zeta_j \bar{\mathbf{d}}_j, \tag{6.45}$$

and

$$\|\mathbf{y}_k\|_{\mathbf{G}}^2 = \sum_{j=1}^{k} \zeta_j^2, \tag{6.46}$$

and we have the error estimate

$$\|\mathbf{y} - \mathbf{y}_k\|_{\mathbf{G}}^2 = \sum_{j=k+1}^{m} \zeta_j^2, \tag{6.47}$$

where $\mathbf{y}$ *is the solution of* (6.36).

Proof. We establish (6.44). The expansion (6.45) follows by repeated application of (6.43). The proof of (6.46) and (6.47) is a direct consequence of (6.44) and (6.45), as in the proof of Theorem 6.4.

We deduce from (6.41) and (6.6) that $\bar{\mathbf{D}}_k^{\mathsf{T}} \mathbf{G} \bar{\mathbf{D}}_k = \bar{\mathbf{R}}_k^{-\mathsf{T}} \mathbf{R}_k^{-\mathsf{T}} \mathbf{V}_k^{\mathsf{T}} \mathbf{G} \mathbf{V}_k \mathbf{R}_k^{-1} \bar{\mathbf{R}}_k^{-1}$. We now expand this expression from the inside and out. Using (4.1), we have

$$\begin{aligned}
(\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A} + \mathbf{N})\mathbf{V}_k &= \mathbf{A}^{\mathsf{T}}\mathbf{U}_{k+1}\mathbf{E}_k + \mathbf{N}\mathbf{V}_k \\
&= \mathbf{N}((\mathbf{V}_k\mathbf{E}_k^{\mathsf{T}} + \alpha_{k+1}\mathbf{v}_{k+1}\mathbf{e}_{k+1}^{\mathsf{T}})\mathbf{E}_k + \mathbf{V}_k) \\
&= \mathbf{N}(\mathbf{V}_k(\mathbf{E}_k^{\mathsf{T}}\mathbf{E}_k + \mathbf{I}_k) + \gamma_k\rho_k\mathbf{v}_{k+1}\mathbf{e}_k^{\mathsf{T}}),
\end{aligned}$$

where we used the identity $\mathbf{e}_{k+1}^{\mathsf{T}}\mathbf{E}_k = \beta_{k+1}\mathbf{e}_k^{\mathsf{T}}$ and the definition (6.40) of γ_k. Using the above identity twice, we obtain, after some basic manipulations,

$$\mathbf{V}_k^{\mathsf{T}}\mathbf{G}\mathbf{V}_k = (\mathbf{E}_k^{\mathsf{T}}\mathbf{E}_k + \mathbf{I}_k)^2 + \gamma_k^2\rho_k^2\mathbf{e}_k\mathbf{e}_k^{\mathsf{T}}.$$

When forming the product above, cross terms vanish because they contain the expression $\mathbf{V}_k^{\mathsf{T}}\mathbf{N}\mathbf{v}_{k+1}$, which is zero by orthogonality.

As we already noticed at the beginning of this section, $\mathbf{E}_k^{\mathsf{T}}\mathbf{E}_k + \mathbf{I}_k = \mathbf{R}_k^{\mathsf{T}}\mathbf{R}_k$, and therefore,

$$(\mathbf{E}_k^{\mathsf{T}}\mathbf{E}_k + \mathbf{I}_k)^2 = \mathbf{R}_k^{\mathsf{T}}\mathbf{R}_k\mathbf{R}_k^{\mathsf{T}}\mathbf{R}_k.$$

Consequently,

$$\mathbf{R}_k^{-\mathsf{T}}\left((\mathbf{E}_k^{\mathsf{T}}\mathbf{E}_k + \mathbf{I}_k)^2 + \gamma_k^2\rho_k^2\mathbf{e}_k\mathbf{e}_k^{\mathsf{T}}\right)\mathbf{R}_k^{-1} = \mathbf{R}_k\mathbf{R}_k^{\mathsf{T}} + \gamma_k^2\mathbf{e}_k\mathbf{e}_k^{\mathsf{T}},$$

where we used the identity $\mathbf{R}_k^{-\mathsf{T}}\mathbf{e}_k = \rho_k^{-1}\mathbf{e}_k$.

Note now that by definition of $\bar{\mathbf{R}}_k$,

$$\bar{\mathbf{R}}_k^{\mathsf{T}}\bar{\mathbf{R}}_k = \begin{bmatrix}\bar{\mathbf{R}}_k^{\mathsf{T}} & \mathbf{0}^{\mathsf{T}}\end{bmatrix}\begin{bmatrix}\bar{\mathbf{R}}_k \\ \mathbf{0}\end{bmatrix} = \begin{bmatrix}\mathbf{R}_k & \gamma_k\mathbf{e}_k\end{bmatrix}\bar{\mathbf{Q}}_k\bar{\mathbf{Q}}_k^{\mathsf{T}}\begin{bmatrix}\mathbf{R}_k^{\mathsf{T}} \\ \gamma_k\mathbf{e}_k^{\mathsf{T}}\end{bmatrix} = \mathbf{R}_k\mathbf{R}_k^{\mathsf{T}} + \gamma_k^2\mathbf{e}_k\mathbf{e}_k^{\mathsf{T}}.$$

This last identity finally yields

$$\bar{\mathbf{D}}_k^{\mathsf{T}}\mathbf{G}\bar{\mathbf{D}}_k = \bar{\mathbf{R}}_k^{-\mathsf{T}}(\mathbf{R}_k\mathbf{R}_k^{\mathsf{T}} + \gamma_k^2\mathbf{e}_k\mathbf{e}_k^{\mathsf{T}})\bar{\mathbf{R}}_k^{-1} = \bar{\mathbf{R}}_k^{-\mathsf{T}}(\bar{\mathbf{R}}_k^{\mathsf{T}}\bar{\mathbf{R}}_k)\bar{\mathbf{R}}_k^{-1} = \mathbf{I}_k,$$

and this completes the proof. □

The last rotation computes $\bar{\mathbf{R}}_k$ by eliminating the subdiagonals of $\mathbf{R}_k^T$, i.e., θ_{j+1} for $j = 1, \ldots, k-1$ as follows:

$$\begin{array}{c} \\ k \\ k+1 \end{array} \begin{array}{c} \begin{array}{cc} k & k+1 \end{array} \\ \begin{bmatrix} \hat{c}_k & \hat{s}_k \\ \hat{s}_k & -\hat{c}_k \end{bmatrix} \end{array} \begin{array}{c} \begin{array}{cc} k & k+1 \end{array} \\ \begin{bmatrix} \tilde{\rho}_k & \\ \theta_{k+1} & \rho_{k+1} \end{bmatrix} \end{array} = \begin{bmatrix} \bar{\rho}_k & \bar{\theta}_{k+1} \\ & \tilde{\rho}_{k+1} \end{bmatrix},$$

with the initialization $\tilde{\rho}_1 := \rho_1$. In other words, $\bar{\rho}_k := \sqrt{\tilde{\rho}_k^2 + \theta_{k+1}^2}$, $\hat{c}_k := \tilde{\rho}_k / \bar{\rho}_k$, $\hat{s}_k := \theta_{k+1}/\bar{\rho}_k$, $\bar{\theta}_{k+1} := \hat{s}_k \rho_{k+1}$ and $\tilde{\rho}_{k+1} := -\hat{c}_k \rho_{k+1}$.

During the k-th iteration, ρ_{k+1} is not yet available. Therefore, the computation of $\bar{\theta}_{k+1}$ and $\tilde{\rho}_{k+1}$ can only take place during the next iteration. The k-th iteration computes $\bar{\theta}_k$ and $\tilde{\rho}_k$ as soon as ρ_k becomes available using the previous values $\hat{c}_{k-1}$ and $\hat{s}_{k-1}$. To update, $\mathbf{y}_k$ we first compute $\bar{\mathbf{d}}_k$ according to (6.42) and update $\mathbf{y}$ using (6.43). Computing $\mathbf{d}_{k+1}$ appears to require ρ_{k+1}. However, as Fong and Saunders (2011) point out, it is possible to bypass this requirement by defining instead

$$\mathbf{h}_k := \rho_k \mathbf{d}_k, \quad \text{and} \quad \bar{\mathbf{h}}_k := \rho_k \bar{\rho}_k \bar{\mathbf{d}}_k,$$

and updating

$$\bar{\mathbf{h}}_k = \mathbf{h}_k + \frac{\rho_k \bar{\theta}_k}{\rho_{k-1} \bar{\rho}_{k-1}} \bar{\mathbf{h}}_{k-1}, \quad \mathbf{y}_k = \mathbf{y}_{k-1} + \frac{\zeta_k}{\rho_{k-1} \bar{\rho}_{k-1}} \bar{\mathbf{h}}_k, \quad \mathbf{h}_{k+1} = \mathbf{v}_{k+1} - \frac{\theta_{k+1}}{\rho_k} \mathbf{h}_k.$$

The least-squares residual is initially given by

$$\begin{bmatrix} \hat{c}_1 & \hat{s}_1 \\ \hat{s}_1 & -\hat{c}_1 \end{bmatrix} \begin{bmatrix} \beta_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \hat{c}_1 \beta_1 \\ \hat{s}_1 \beta_1 \end{bmatrix},$$

so that $\zeta_1 := \hat{c}_1 \beta_1$. Define $\hat{\zeta}_2 := \hat{s}_1 \beta_1$ to obtain the recursion

$$\begin{bmatrix} \hat{c}_k & \hat{s}_k \\ \hat{s}_k & -\hat{c}_k \end{bmatrix} \begin{bmatrix} \hat{\zeta}_k \\ 0 \end{bmatrix} = \begin{bmatrix} \hat{c}_k \hat{\zeta}_k \\ \hat{s}_k \hat{\zeta}_k \end{bmatrix},$$

i.e., $\zeta_k := \hat{c}_k \hat{\zeta}_k$ and $\hat{\zeta}_{k+1} := \hat{s}_k \hat{\zeta}_k$. The recursion begins with $\hat{\zeta}_1 := \zeta_1$. Note that there is a lag in the residual value—the component $\hat{\zeta}_{k+1}$ is obtained when computing $\mathbf{x}_k$ but it is $|\zeta_k|$ that gives the residual value, and it corresponds to $\mathbf{x}_{k-1}$, not to $\mathbf{x}_k$. The main computational steps of G-LSMR are summarized as Algorithm 6.3.

6.7. **Generalized CRAIG-MR.** In this section, we present the main features of a method similar to LSMR but that is equivalent to applying MINRES to the Schur-complement equations. By analogy with LSMR, we dub this method CRAIG-MR. Its application to (6.16) is dubbed the *generalized* CRAIG-MR, or G-CRAIG-MR. The reason for introducing this method becomes clear in §8.4, where we show that MINRES applied directly to (1.1) with right-hand side $(\mathbf{b}, \mathbf{0})$ alternates between G-LSMR steps and G-CRAIG-MR steps.

It follows from (6.19) that the approximation $\mathbf{x}_k = \mathbf{U}_k \bar{\mathbf{x}}_k$ reveals the associated Lanczos process

$$\begin{aligned} (\mathbf{M} + \mathbf{A}\mathbf{N}^{-1}\mathbf{A}^T)\mathbf{U}_k &= \mathbf{M}\mathbf{U}_k(\mathbf{B}_k\mathbf{B}_k^T + \mathbf{I}_k) + \alpha_k\beta_{k+1}\mathbf{M}\mathbf{u}_{k+1}\mathbf{e}_k^T, \\ &= \mathbf{M}\mathbf{U}_{k+1} \begin{bmatrix} \mathbf{B}_k\mathbf{B}_k^T + \mathbf{I}_k \\ \alpha_k\beta_{k+1}\mathbf{e}_k^T \end{bmatrix}, \end{aligned} \tag{6.48}$$

Algorithm 6.3 Generalized LSMR

Require: $\mathbf{M}$, $\mathbf{A}$, $\mathbf{N}$, $\mathbf{b}$, d, τ, k_{max}
$\beta_1\mathbf{M}\mathbf{u}_1 = \mathbf{b}, \quad \alpha_1\mathbf{N}\mathbf{v}_1 = \mathbf{A}^\mathsf{T}\mathbf{u}_1$ *// Initialize bidiagonalization*
$\hat{\alpha}_1 = \alpha_1, \quad \bar{\rho}_1 = \alpha_1, \quad \hat{\zeta}_1 = \alpha_1\beta_1, \quad \hat{c}_0 = -1, \quad \hat{s}_0 = 0, \quad \rho_0 = 0, \quad \bar{\rho}_0 = 0$
$\mathbf{h}_1 = \mathbf{v}_1, \quad \bar{\mathbf{h}}_0 = \mathbf{0}, \quad \mathbf{y}_0 = \mathbf{0}$
$k = 1, \quad \Delta = 0, \quad$ converged $=$ **false**
while not converged **and** $k < k_{max}$ **do**
  // Continue bidiagonalization
  $\beta_{k+1}\mathbf{M}\mathbf{u}_{k+1} = \mathbf{A}\mathbf{v}_k - \alpha_k\mathbf{M}\mathbf{u}_k, \quad \alpha_{k+1}\mathbf{N}\mathbf{v}_{k+1} = \mathbf{A}^\mathsf{T}\mathbf{u}_{k+1} - \beta_{k+1}\mathbf{N}\mathbf{v}_k$
  $\hat{\rho}_k = (1 + \hat{\alpha}_k^2)^{\frac{1}{2}}, \quad c_k = \hat{\alpha}_k/\hat{\rho}_k, \quad s_k = 1/\hat{\rho}_k$ *// Rotation of type II*
  $\rho_k = (\hat{\rho}_k^2 + \beta_{k+1}^2)^{\frac{1}{2}}, \quad \bar{c}_k = \hat{\rho}_k/\rho_k, \quad \bar{s}_k = \beta_{k+1}/\rho_k$ *// Rotation of type I*
  $\theta_{k+1} = \bar{s}_k\alpha_{k+1}, \quad \hat{\alpha}_{k+1} = -\bar{c}_k\alpha_{k+1},$
  $\bar{\theta}_k = \hat{s}_{k-1}\rho_k, \quad \tilde{\rho}_k = -\hat{c}_{k-1}\rho_k$ *// Rotation of type III*
  $\bar{\rho}_k = (\tilde{\rho}_k^2 + \theta_{k+1}^2)^{\frac{1}{2}}, \quad \hat{c}_k = \tilde{\rho}_k/\bar{\rho}_k, \quad \hat{s}_k = \theta_{k+1}/\bar{\rho}_k,$
  $\zeta_k = \hat{c}_k\hat{\zeta}_k, \quad \hat{\zeta}_{k+1} = \hat{s}_k\hat{\zeta}_k, \quad \Delta = \Delta + \zeta_k^2$ *// Residual and error update*
  $\bar{\mathbf{h}}_k = \mathbf{h}_k + \rho_k\bar{\theta}_k/(\rho_{k-1}\bar{\rho}_{k-1})\,\bar{\mathbf{h}}_{k-1}$ *// Update*
  $\mathbf{y}_k = \mathbf{y}_{k-1} + \zeta_k/(\rho_{k-1}\bar{\rho}_{k-1})\,\bar{\mathbf{h}}_k$
  $\mathbf{h}_{k+1} = \mathbf{v}_{k+1} - \theta_{k+1}/\rho_k\,\mathbf{h}_k$
  if $k \geq d$ **then**
    converged $= \left(\sum_{j=k-d+1}^{k}\zeta_j^2 < \tau^2\Delta\right)$ *// Test convergence*
  $k \leftarrow k + 1$
$\mathbf{y} = \mathbf{y}_k$
$\mathbf{x} = \mathbf{M}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{y})$
return $(\mathbf{x}, \mathbf{y})$

where we note that the matrix $(\mathbf{B}_k\mathbf{B}_k^\mathsf{T} + \mathbf{I}_k)$ is tridiagonal, and by definition, $\mathbf{b} = \beta_1\mathbf{M}\mathbf{u}_1$.

Note that for any $\mathbf{x}$, the residual of (6.16) lies in $\mathbb{M}^\star$. The appropriate norm to measure this residual is thus the $\mathbf{M}^{-1}$-norm. By definition, G-CRAIG-MR computes $\mathbf{x}_k$ as the solution of

$$\underset{\mathbf{x}\in\mathrm{Range}(\mathbf{V}_k)}{\text{minimize}} \quad \tfrac{1}{2}\|\mathbf{M}^{-\frac{1}{2}}(\mathbf{b} - (\mathbf{M} + \mathbf{A}\mathbf{N}^{-1}\mathbf{A}^\mathsf{T})\mathbf{x})\|_2.$$

Since the system is always consistent in $\mathbb{R}^n$, the final solution $\mathbf{x}$ solves

$$\mathbf{M}^{-\frac{1}{2}}(\mathbf{M} + \mathbf{A}\mathbf{N}^{-1}\mathbf{A}^\mathsf{T})\mathbf{x} = \mathbf{M}^{-\frac{1}{2}}\mathbf{b}. \tag{6.49}$$

We have established the following result, which follows directly from the very definition of G-CRAIG-MR and the Lanczos process (6.48).

Theorem 6.7. *The generalized* CRAIG-MR *iterates on (2.4) are the same as those generated by the* MINRES *method on the positive definite system (2.7) in the metric defined by* $\mathbf{M}$.

Using the approximation $\mathbf{x} \approx \mathbf{x}_k = \mathbf{U}_k \bar{\mathbf{x}}_k$ and (6.48), the k-th subproblem may be written

$$\underset{\bar{\mathbf{x}}_k}{\text{minimize}} \ \tfrac{1}{2} \left\| \beta_1 \mathbf{e}_1 - \begin{bmatrix} \mathbf{B}_k \mathbf{B}_k^T + \mathbf{I}_k \\ \alpha_k \beta_{k+1} \mathbf{e}_k^T \end{bmatrix} \bar{\mathbf{x}}_k \right\|_2, \tag{6.50}$$

which is again a regularized subproblem with regularization parameter $\lambda = 1$.

An implementation of G-Craig-mr starts, as in G-Craig, by computing the LQ factorization (6.14), where $\hat{\mathbf{B}}_k$ is k-by-k lower bidiagonal. As before, $\mathbf{B}_k \mathbf{B}_k^T + \mathbf{I}_k = \hat{\mathbf{B}}_k \hat{\mathbf{B}}_k^T$. We define

$$\mathbf{t}_k := \hat{\mathbf{B}}_k^T \bar{\mathbf{x}}_k, \quad \text{and} \quad \mathbf{q}_k := \hat{\mathbf{B}}_k^{-1} (\alpha_k \beta_{k+1} \mathbf{e}_k) = \frac{\alpha_k \beta_{k+1}}{\hat{\alpha}_k} \mathbf{e}_k := \gamma_k \mathbf{e}_k.$$

The least-squares residual of the k-th subproblem may now be rewritten

$$\beta_1 \mathbf{e}_1 - \begin{bmatrix} \mathbf{B}_k \mathbf{B}_k^T + \mathbf{I}_k \\ \alpha_k \beta_{k+1} \mathbf{e}_k^T \end{bmatrix} \bar{\mathbf{x}}_k = \beta_1 \mathbf{e}_1 - \begin{bmatrix} \hat{\mathbf{B}}_k \hat{\mathbf{B}}_k^T \\ \mathbf{q}_k^T \hat{\mathbf{B}}_k^T \end{bmatrix} \bar{\mathbf{x}}_k = \beta_1 \mathbf{e}_1 - \begin{bmatrix} \hat{\mathbf{B}}_k \\ \gamma_k \mathbf{e}_k^T \end{bmatrix} \mathbf{t}_k.$$

As in Lsmr, we now perform the third QR factorization

$$\bar{\mathbf{Q}}_{k+1} \begin{bmatrix} \hat{\mathbf{B}}_k & \beta_1 \mathbf{e}_1 \\ \gamma_k \mathbf{e}_k^T & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_k & \mathbf{z}_k \\ \mathbf{0} & \zeta_{k+1} \end{bmatrix} \quad \text{with} \quad \mathbf{R}_k = \begin{bmatrix} \rho_1 & \theta_2 & & \\ & \rho_2 & \ddots & \\ & & \ddots & \theta_k \\ & & & \rho_k \end{bmatrix},$$

where $\bar{\mathbf{Q}}_{k+1}$ is orthogonal. Again for this second factorization to be well defined recursively, it is necessary to verify that $\gamma_k = \hat{\beta}_{k+1}$. Fortunately, we have already verified this fact in (6.24). This helps us write the k-th subproblem as

$$\underset{\mathbf{t}_k}{\text{minimize}} \ \tfrac{1}{2} \left\| \begin{bmatrix} \mathbf{z}_k \\ \zeta_{k+1} \end{bmatrix} - \begin{bmatrix} \mathbf{R}_k \\ \mathbf{0} \end{bmatrix} \mathbf{t}_k \right\|_2.$$

Clearly, we obtain the solution by picking $\mathbf{t}_k := \mathbf{R}_k^{-1} \mathbf{z}_k$ and the least-squares residual is simply $|\zeta_{k+1}|$. The remaining algorithmic details essentially follow those developed by Fong and Saunders (2011) for Lsmr.

Seeking once again a direct update of $\mathbf{x}_k$, we define the columns $\mathbf{d}_1$ to $\mathbf{d}_k$ of $\mathbf{D}_k$ recursively via the definition

$$\mathbf{D}_k := \mathbf{U}_k \hat{\mathbf{B}}_k^{-T}, \qquad \text{i.e.,} \qquad \hat{\mathbf{B}}_k \mathbf{D}_k^T = \mathbf{U}_k^T,$$

which yields

$$\mathbf{d}_1 := \frac{1}{\hat{\alpha}_1} \mathbf{u}_1, \qquad \text{and} \qquad \mathbf{d}_{j+1} := \frac{1}{\hat{\alpha}_{j+1}} (\mathbf{u}_{j+1} - \hat{\beta}_{j+1} \mathbf{d}_j) \quad (j \geq 0).$$

Similarly, define the columns $\bar{\mathbf{d}}_1$ to $\bar{\mathbf{d}}_k$ of $\bar{\mathbf{D}}_k$ via

$$\bar{\mathbf{D}}_k := \mathbf{D}_k \mathbf{R}_k^{-1}, \qquad \text{i.e.,} \qquad \mathbf{R}_k^T \bar{\mathbf{D}}_k^T = \mathbf{D}_k^T, \tag{6.51}$$

which yields

$$\bar{\mathbf{d}}_1 := \frac{1}{\rho_1} \mathbf{d}_1, \qquad \text{and} \qquad \bar{\mathbf{d}}_{j+1} := \frac{1}{\rho_{j+1}} (\mathbf{d}_{j+1} - \theta_{j+1} \bar{\mathbf{d}}_j), \quad (j \geq 0).$$

With these definitions, we may use the update

$$\mathbf{x}_k = \mathbf{U}_k \bar{\mathbf{x}}_k = \mathbf{U}_k \hat{\mathbf{B}}_k^{-T} \mathbf{t}_k = \mathbf{D}_k \mathbf{t}_k = \mathbf{D}_k \mathbf{R}_k^{-1} \mathbf{z}_k = \bar{\mathbf{D}}_k \mathbf{z}_k = \mathbf{x}_{k-1} + \zeta_k \bar{\mathbf{d}}_k.$$

Theorem 6.8. *Let $\bar{\mathbf{D}}_k$ be defined as in (6.51). Then, for $k = 1, \dots, n$, we have*

$$\bar{\mathbf{D}}_k^{\mathsf{T}} \mathbf{W} \bar{\mathbf{D}}_k = \mathbf{I}_k, \quad \text{where } \mathbf{W} := (\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}} + \mathbf{M})\mathbf{M}^{-1}(\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}} + \mathbf{M}). \tag{6.52}$$

In particular,

$$\mathbf{x}_k = \sum_{j=1}^{k} \zeta_j \bar{\mathbf{d}}_j, \tag{6.53}$$

and

$$\|\mathbf{x}_k\|_{\mathbf{W}}^2 = \sum_{j=1}^{k} \zeta_j^2, \tag{6.54}$$

and we have the error estimate

$$\|\mathbf{x} - \mathbf{x}_k\|_{\mathbf{W}}^2 = \sum_{j=k+1}^{n} \zeta_j^2, \tag{6.55}$$

where $\mathbf{x}$ is the solution of (6.49).

Proof. The proof is analogous to that of Theorem 6.6. □

The rotation necessary to compute $\mathbf{R}_k$, which we call a rotation of type III, eliminates the subdiagonals of $\hat{\mathbf{B}}_k$, i.e., $\hat{\beta}_{j+1}$, $j = 1, \dots, k-1$, as follows:

$$\begin{array}{c} \\ k \\ k+1 \end{array} \begin{array}{c} \begin{array}{cc} k & k+1 \end{array} \\ \begin{bmatrix} \hat{c}_k & \hat{s}_k \\ \hat{s}_k & -\hat{c}_k \end{bmatrix} \end{array} \begin{array}{c} \begin{array}{cc} k & k+1 \end{array} \\ \begin{bmatrix} \tilde{\alpha}_k & \\ \hat{\beta}_{k+1} & \hat{\alpha}_{k+1} \end{bmatrix} \end{array} = \begin{bmatrix} \rho_k & \theta_{k+1} \\ & \tilde{\alpha}_{k+1} \end{bmatrix},$$

with the initialization $\tilde{\alpha}_1 := \hat{\alpha}_1$. In other words, $\rho_k := \sqrt{\tilde{\alpha}_k^2 + \hat{\beta}_{k+1}^2}$, $\hat{c}_k := \tilde{\alpha}_k / \rho_k$, $\hat{s}_k := \hat{\beta}_{k+1}/\rho_k$, $\theta_{k+1} := \hat{s}_k \hat{\alpha}_{k+1}$, and $\tilde{\alpha}_{k+1} = -\hat{c}_k \hat{\alpha}_{k+1}$.

The main details of G-Craig-mr are summarized as Algorithm 6.4. It is worth noting again that the least-squares residual, $|\zeta_k|$ lags one step behind and corresponds to the *previous* iterate $\mathbf{x}_{k-1}$.

7. Upper Bound Error Estimates

In the previous sections, we proposed several lower bounds on the direct error that are linked to the techniques described by Golub and Meurant (2010). Even though those lower bounds estimate the error at step $k-d$, it is safe to use the most recent solution estimate, i.e., that computed at step k, owing to the monotonicity of the error sequence.

To motivate our approach, consider the generalized Lsqr method and the relation (6.5), which determines the coefficients $\bar{\mathbf{y}}_k$ of the k-th approximation $\mathbf{y}_k$ in terms of the initial values $\beta_1 = \|\mathbf{b}\|_{\mathbf{M}^{-1}}$ and $\alpha_1 = \|\mathbf{A}^{\mathsf{T}}\mathbf{u}_1\|_{\mathbf{N}^{-1}} = \beta_1^{-1}\|\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{b}\|_{\mathbf{N}^{-1}}$. Let $\hat{\mathbf{T}}_k$ denote the symmetric and positive definite tridiagonal matrix $\mathbf{R}_k^{\mathsf{T}}\mathbf{R}_k$. Then $\bar{\mathbf{y}}_k = \alpha_1\beta_1\hat{\mathbf{T}}_k^{-1}\mathbf{e}_1$. Theorem 6.2 indicates that $\|\mathbf{y}_k\|_{\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A}+\mathbf{N}} = \|\mathbf{z}_k\|_2$ where $\mathbf{z}_k = \mathbf{R}_k\bar{\mathbf{y}}_k$. Therefore,

$$\|\mathbf{y}_k\|_{\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A}+\mathbf{N}} = \|\mathbf{z}_k\|_2 = \|\mathbf{R}_k\bar{\mathbf{y}}_k\|_2 = \|\bar{\mathbf{y}}_k\|_{\hat{\mathbf{T}}_k} = |\alpha_1\beta_1|\,\|\mathbf{e}_1\|_{\hat{\mathbf{T}}_k^{-1}}.$$

Algorithm 6.4 Generalized CRAIG-MR

Require: $\mathbf{M}$, $\mathbf{A}$, $\mathbf{N}$, $\mathbf{b}$, d, τ, $k_{\max}$
$\beta_1\mathbf{M}\mathbf{u}_1 = \mathbf{b}, \quad \alpha_1\mathbf{N}\mathbf{v}_1 = \mathbf{A}^{\mathsf{T}}\mathbf{u}_1$ *// Initialize bidiagonalization*
$\delta_1 = 1, \quad \hat{\alpha}_1 = (\alpha_1^2 + 1)^{\frac{1}{2}}, \quad c_1 = \alpha_1/\hat{\alpha}_1, \quad s_1 = 1/\hat{\alpha}_1$
$\hat{\zeta}_1 = \beta_1, \quad \tilde{\alpha}_1 = \hat{\alpha}_1, \quad \theta_1 = 0$
$\mathbf{d}_1 = 1/\hat{\alpha}_1\mathbf{u}_1, \quad \bar{\mathbf{d}}_0 = \mathbf{0}, \quad \mathbf{x}_0 = \mathbf{0}$
$k = 1, \quad \Delta = 0, \quad$ converged $=$ **false**
while not converged **and** $k < k_{max}$ **do**
  // Continue bidiagonalization
  $\beta_{k+1}\mathbf{M}\mathbf{u}_{k+1} = \mathbf{A}\mathbf{v}_k - \alpha_k\mathbf{M}\mathbf{u}_k, \quad \alpha_{k+1}\mathbf{N}\mathbf{v}_{k+1} = \mathbf{A}^{\mathsf{T}}\mathbf{u}_{k+1} - \beta_{k+1}\mathbf{N}\mathbf{v}_k$
  $\hat{\beta}_{k+1} = c_k\beta_{k+1}, \quad \gamma_{k+1} = s_k\beta_{k+1}$ *// Continue rotation of type I*
  $\delta_{k+1} = (\gamma_{k+1}^2 + 1)^{\frac{1}{2}}, \ \bar{c}_k = -1/\delta_{k+1}, \ \bar{s}_k = \gamma_{k+1}/\delta_{k+1}$ *// Rotation of type II*
  // Compute new Givens rotation of type I
  $\hat{\alpha}_{k+1} = (\alpha_{k+1}^2 + \delta_{k+1}^2)^{\frac{1}{2}}, \quad c_{k+1} = \alpha_{k+1}/\hat{\alpha}_{k+1}, \quad s_{k+1} = \delta_{k+1}/\hat{\alpha}_{k+1}$
  $\rho_k = (\tilde{\alpha}_k^2 + \hat{\beta}_{k+1}^2)^{\frac{1}{2}}, \quad \hat{c}_k = \tilde{\alpha}_k/\rho_k, \quad \hat{s}_k = \hat{\beta}_{k+1}/\rho_k$ *// Rotation of type III*
  $\theta_{k+1} = \hat{s}_k\hat{\alpha}_{k+1}, \quad \tilde{\alpha}_{k+1} = -\hat{c}_k\hat{\alpha}_{k+1}$
  $\zeta_k = \hat{c}_k\hat{\zeta}_k, \quad \hat{\zeta}_{k+1} = \hat{s}_k\hat{\zeta}_k, \quad \Delta = \Delta + \zeta_k^2$ *// Update*
  $\mathbf{d}_{k+1} = 1/\hat{\alpha}_{k+1}(\mathbf{u}_{k+1} - \hat{\beta}_{k+1}\mathbf{d}_k)$
  $\bar{\mathbf{d}}_k = 1/\rho_k(\mathbf{d}_k - \theta_k\bar{\mathbf{d}}_{k-1})$
  $\mathbf{x}_k = \mathbf{x}_{k-1} + \zeta_k\bar{\mathbf{d}}_k$
  if $k \geq d$ **then**
    converged $= \left(\sum_{j=k-d+1}^{k}\zeta_j^2 < \tau^2\Delta\right)$ *// Test convergence*
  $k \leftarrow k + 1$
$\mathbf{x} = \mathbf{x}_k$
$\mathbf{y} = \mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}}\mathbf{x}$
return $(\mathbf{x}, \mathbf{y})$

But $\|\mathbf{e}_1\|^2_{\hat{\mathbf{T}}_k^{-1}} = \mathbf{e}_1^{\mathsf{T}}\hat{\mathbf{T}}_k^{-1}\mathbf{e}_1$ and so the above states that the squared energy norm of $\mathbf{y}_k$ is a factor of the leading element of $\hat{\mathbf{T}}_k^{-1}$. Using the same logic as in the proof of Theorem 6.2, if $\hat{\mathbf{T}} := \hat{\mathbf{T}}_n$ is the value of the tridiagonal when convergence has occurred, then we may measure the direct error as

$$\|\mathbf{y}_k - \mathbf{y}\|_{\mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A}+\mathbf{N}} = |\alpha_1\beta_1| \left| \|\mathbf{e}_1\|_{\mathbf{T}_k^{-1}} - \|\mathbf{e}_1\|_{\mathbf{T}^{-1}} \right|,$$

and this direct error is related to the quality of the approximation of the leading entry of $\hat{\mathbf{T}}^{-1}$ at step k. The same reasoning holds for the generalized LSMR method using (6.39). A similar result follows for the generalized CRAIG and CRAIG-MR methods using (6.17) and (6.50), with the difference that α_1 does not appear explicitly.

In order for the discussion of this section to apply to all four methods, we introduce the notation

$$\hat{\mathbf{T}}_k := \hat{\mathbf{U}}_k^{\mathsf{T}}\hat{\mathbf{U}}_k,$$

where the value of $\hat{\mathbf{U}}_k$ is given in Table 1 for each method based on reduced equations. Note that $\hat{\mathbf{U}}_k$ is upper bidiagonal and $\hat{\mathbf{T}}_k$ is tridiagonal, symmetric and positive definite. From theorems 6.2, 6.4, 6.6, and 6.8 the error estimates are related to the approximation of the leading entry $\hat{\mathbf{T}}_{11}^{-1}$ by $(\hat{\mathbf{T}}_k^{-1})_{11}$.

TABLE 1. Factor $\hat{\mathbf{U}}_k$ of the tridiagonal and coefficient matrix $\mathbf{W}$ of the system solved by each method based on a reduced system. The same matrix $\mathbf{W}$ defines the energy norm for the method.

Method	$\hat{\mathbf{U}}_k$	$\mathbf{W}$	Theorems
G-LSQR	$\mathbf{R}_k$	$\mathbf{A}^T\mathbf{M}^{-1}\mathbf{A}+\mathbf{N}$	6.1 and 6.2
G-CRAIG	$\hat{\mathbf{B}}_k^T$	$\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^T+\mathbf{M}$	6.3 and 6.4
G-LSMR	$\mathbf{R}_k$	$(\mathbf{A}^T\mathbf{M}^{-1}\mathbf{A}+\mathbf{N})\mathbf{N}^{-1}(\mathbf{A}^T\mathbf{M}^{-1}\mathbf{A}+\mathbf{N})$	6.5 and 6.6
G-CRAIG-MR	$\hat{\mathbf{B}}_k^T$	$(\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^T+\mathbf{M})\mathbf{M}^{-1}(\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^T+\mathbf{M})$	6.7 and 6.8

The vector $\mathbf{z}$ of entries ζ_j defines different quantities depending on the algorithm choice. In G-LSQR and G-LSMR,

(7.1)
$$\|\mathbf{z}\|_2^2 = \|\mathbf{A}^T\mathbf{M}^{-1}\mathbf{b}\|_{\mathbf{N}^{-1}}^2\ \hat{\mathbf{T}}_{11}^{-1} \quad \text{and} \quad \sum_{j=k+1}^{n} \zeta_j^2 = \|\mathbf{A}^T\mathbf{M}^{-1}\mathbf{b}\|_{\mathbf{N}^{-1}}^2\ \left(\hat{\mathbf{T}}_{11}^{-1} - (\hat{\mathbf{T}}_k^{-1})_{11}\right),$$

while in G-CRAIG and G-CRAIG-MR,

$$\|\mathbf{z}\|_2^2 = \|\mathbf{b}\|_{\mathbf{M}^{-1}}^2\ \hat{\mathbf{T}}_{11}^{-1} \quad \text{and} \quad \sum_{j=k+1}^{m} \zeta_j^2 = \|\mathbf{b}\|_{\mathbf{M}^{-1}}^2\ \left(\hat{\mathbf{T}}_{11}^{-1} - (\hat{\mathbf{T}}_k^{-1})_{11}\right), \tag{7.2}$$

where $\hat{\mathbf{T}}_k$ is the $k \times k$ principal submatrix of $\hat{\mathbf{T}}$.

Let $0 < \lambda_1 \leq \cdots \leq \lambda_p$ be the eigenvalues of the matrix $\mathbf{W}$ defining the energy norm in the method chosen, where p is either m or n. As explained in Theorems 6.2, 6.4, 6.6 and 6.8 and summarized in Table 1, $\mathbf{W}$ is also the coefficient matrix of the system being solved. Therefore, $\mathbf{W}$ and $\hat{\mathbf{T}}$ have the same eigenvalues.

Let $\hat{\mathbf{T}} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$ be the eigendecomposition of $\hat{\mathbf{T}}$, where $\mathbf{Q}$ is orthogonal and $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_p)$. The squared energy norm of the solution may then be expressed as

$$\|\mathbf{z}\|_2^2 = \gamma^2\, \mathbf{e}_1^T \hat{\mathbf{T}}^{-1} \mathbf{e}_1 = \gamma^2\, \boldsymbol{\mu}^T \mathbf{\Lambda}^{-1} \boldsymbol{\mu} = \gamma^2 \sum_{i=1}^{p} \lambda_i^{-1} \mu_i^2, \tag{7.3}$$

where $\gamma > 0$ is either $\|\mathbf{A}^T\mathbf{M}^{-1}\mathbf{b}\|_{\mathbf{N}^{-1}}$ or $\|\mathbf{b}\|_{\mathbf{M}^{-1}}$ and $\boldsymbol{\mu} := \mathbf{Q}^T\mathbf{e}_1 = (\mu_1, \ldots, \mu_p)$. Note that the components of $\boldsymbol{\mu}$ are the first components of the normalized eigenvectors of $\hat{\mathbf{T}}$. Similarly, the squared energy norm of the k-th approximation may be written as

$$\|\mathbf{z}_k\|_2^2 = \gamma^2 \sum_{i=1}^{k} \lambda_i^{-1} \mu_i^2,$$

so that the squared energy norm of the error is given by

$$\|\mathbf{z} - \mathbf{z}_k\|_2^2 = \gamma^2 \sum_{i=k+1}^{p} \lambda_i^{-1} \mu_i^2.$$

As earlier, we may choose to terminate the iterations as soon as

$$\sum_{i=k-d+1}^{k} \lambda_i^{-1} \mu_i^2 \leq \tau^2 \sum_{i=1}^{k} \lambda_i^{-1} \mu_i^2 \tag{7.4}$$

for a given window size $d \in \mathbb{N}_0$ and tolerance $\tau^2 > 0$. Note that this relative stopping test does not depend on γ.

Following the exposition of Golub and Meurant (1997) and Golub and Meurant (2010), the energy norm of $\mathbf{z}$ can be interpreted as the approximation by a Gauss quadrature of the Riemann-Stieltjes integral

$$\int_{\lambda_1}^{\lambda_p} \frac{1}{\lambda} \, \mathrm{d}\mu(\lambda), \tag{7.5}$$

where the measure μ is the nondecreasing step function

$$\mu(\lambda) = \begin{cases} 0 & \text{if } \lambda < \lambda_1, \\ \sum_{i=1}^{k} \mu_i^2 & \text{if } \lambda_k \le \lambda < \lambda_{k+1}, \\ \sum_{i=1}^{p} \mu_i^2 & \text{if } \lambda \ge \lambda_p. \end{cases}$$

Comparing the last sum in (7.3) with (7.5), we see that in this Gauss approximation, the nodes are given by the eigenvalues of $\hat{\mathbf{T}}$ while the weights are the squared first components of the normalized eigenvectors of $\hat{\mathbf{T}}$. Following this interpretation, the errors (7.1) and (7.2) can be viewed as the remainder of the approximation of (7.5) by this Gauss quadrature. In practice there is no need to compute explicitly the eigenvalues λ_i or the weights μ_i—we simply accumulate the tems of the quadrature by computing recursively $\mathbf{e}_1^{\mathsf{T}} \hat{\mathbf{T}}_k^{-1} \mathbf{e}_1$ for all k, stopping when new terms no longer add significantly to the overall sum.

The interesting feature of the above interpretation, as noted by Golub and Meurant (2010), is that because the sign of the remainder can be known in advance, the quadrature approximation will either yield an upper or a lower bound on the energy norm. This is due to the sign of the derivatives of $\lambda \mapsto 1/\lambda$ being known in advance. A pure Gauss approximation can be shown to yield a lower bound. But other quadratures are possible, such as the Gauss-Radau quadrature in which one node is fixed, or the Gauss-Lobatto quadrature in which two nodes are fixed. Fixing nodes amounts to augmenting $\hat{\mathbf{T}}_k$ so as to give it one or two prescribed eigenvalues. It can be shown that the Gauss-Radau rule yields an upper bound if $a \le \lambda_1$ is the fixed node, or a lower bound if $b \ge \lambda_p$ is the fixed node, while the Gauss-Lobatto rule, in which both $a \le \lambda_1$ and $b \ge \lambda_p$ are fixed nodes, yields an upper bound. Note that the measure $\mu(\lambda)$ ensures that

$$\int_a^b \frac{1}{\lambda} \, \mathrm{d}\mu(\lambda) = \int_{\lambda_1}^{\lambda_p} \frac{1}{\lambda} \, \mathrm{d}\mu(\lambda)$$

for all such values of a and b.

We denote by ς_j the diagonal entries of $\hat{\mathbf{U}}$ and ν_j the entries under the diagonal, so that

$$\hat{\mathbf{T}}_k = \begin{bmatrix} \varsigma_1 & & & \\ \nu_2 & \varsigma_2 & & \\ & \ddots & \ddots & \\ & & \nu_k & \varsigma_k \end{bmatrix} \begin{bmatrix} \varsigma_1 & \nu_2 & & \\ & \varsigma_2 & \ddots & \\ & & \ddots & \nu_k \\ & & & \varsigma_k \end{bmatrix} = \begin{bmatrix} \varsigma_1^2 & \varsigma_1\nu_2 & & & \\ \varsigma_1\nu_2 & \varsigma_2^2 + \nu_2^2 & \ddots & & \\ & \ddots & \ddots & \varsigma_{k-1}\nu_k & \\ & & \varsigma_{k-1}\nu_k & \varsigma_k^2 + \nu_k^2 \end{bmatrix}.$$

Note that in all four methods, $\hat{\mathbf{T}}_k$ has the form $\mathbf{E}_k^{\mathsf{T}}\mathbf{E}_k + \mathbf{I}_k$ or $\mathbf{B}_k\mathbf{B}_k^{\mathsf{T}} + \mathbf{I}_k$. In both cases, $\lambda_1 \ge 1$. Let $0 < a < 1$ be a lower bound on all the eigenvalues of $\hat{\mathbf{T}}$. We now follow Golub and Meurant (2010) and describe how to implement the

Gauss-Radau rule with a fixed node at a, thereby obtaining an upper bound on the error. When advancing from $\hat{\mathbf{T}}_k$ to $\hat{\mathbf{T}}_{k+1}$, we compute the new off-diagonal element $\varsigma_k\nu_{k+1}$ and modify the new diagonal element so that $\hat{\mathbf{T}}_{k+1}$ has an eigenvalue equal to a. The key element to setting the appropriate value on the diagonal resides in the relationship between the Lanczos process and the family of normalized polynomials $p_i(\lambda)$ that are orthogonal with respect to the measure μ, i.e.,

$$\int_{\lambda_1}^{\lambda_p} p_i(\lambda)p_j(\lambda)\mathrm{d}\mu(\lambda) = \delta_{ij},$$

where $\delta_{ij} = 1$ if $i = j$ and zero otherwise is the Kronecker symbol. Like all orthogonal polynomials, these satsify a three-term recurrence relationship that is none other than, in matrix form,

$$\lambda\mathbf{p}_{k+1}(\lambda) = \hat{\mathbf{T}}_{k+1}\mathbf{p}_{k+1}(\lambda) + \varsigma_{k+1}\nu_{k+2}p_{k+1}(\lambda)\mathbf{e}_{k+1}, \tag{7.6}$$

where $\mathbf{p}_{k+1}(\lambda) = (p_0(\lambda), \dots, p_k(\lambda))$. From this relation, it becomes apparent that $\lambda = a$ is an eigenvalue of $\hat{\mathbf{T}}_{k+1}$ if and only if $p_{k+1}(a) = 0$. In this case, the last equation of (7.6) reads

$$ap_k(a) = \varsigma_k\nu_{k+1}p_{k-1}(a) + (\varsigma_{k+1}^2 + \nu_{k+1}^2)p_k(a).$$

We now modify the $(\varsigma_{k+1}^2+\nu_{k+1}^2)$ on the diagonal and replace it with the value that ensures satisfaction of this last identity, i.e.,

$$\omega_{k+1} := a - \varsigma_k\nu_{k+1}\frac{p_{k-1}(a)}{p_k(a)}. \tag{7.7}$$

The modified tridiagonal may be written as

$$\tilde{\mathbf{T}}_{k+1} = \begin{bmatrix} \hat{\mathbf{T}}_k & \varsigma_k\nu_{k+1}\mathbf{e}_k \\ \varsigma_k\nu_{k+1}\mathbf{e}_k^T & \omega_{k+1} \end{bmatrix}. \tag{7.8}$$

By construction, the smallest eigenvalue of $\tilde{\mathbf{T}}_{k+1}$ is precisely a. The next difficulty is that the polynomials $p_i(\lambda)$ are not directly accessible. Fortunately, it is possible to evaluate ω_{k+1} by extracting the k-th component δ_k of the solution $\boldsymbol{\delta}_k$ of a symmetric, positive-definite and tridiagonal system. To see this, note that (7.6) at iteration k evaluated at $\lambda = a$ may be written

$$(\hat{\mathbf{T}}_k - a\mathbf{I}_k)\boldsymbol{\delta}_k = -\varsigma_k^2\nu_{k+1}^2\mathbf{e}_k, \tag{7.9}$$

where

$$\boldsymbol{\delta}_k = (\delta_1, \dots, \delta_k) := \frac{\varsigma_k\nu_{k+1}}{p_k(a)}\,\mathbf{p}_k(a) = \varsigma_k\nu_{k+1}\left(\frac{p_0(a)}{p_k(a)}, \dots, \frac{p_{k-1}(a)}{p_k(a)}\right).$$

Therefore, (7.7) may equivalently be written

$$\omega_{k+1} = a + \delta_k.$$

Analogously to Arioli (2010) and Golub and Meurant (2010), we can recursively compute δ_k and ω_{k+1} by using the Cholesky decomposition for the system (7.9).

Let

$$\hat{\mathbf{T}}_k - a\mathbf{I}_k = \begin{bmatrix} \varsigma_1^2 - a & \varsigma_1\nu_2 & & & \\ \varsigma_1\nu_2 & \varsigma_2^2 + \nu_2^2 - a & \ddots & & \\ & \ddots & \ddots & \varsigma_{k-1}\nu_k & \\ & & \varsigma_{k-1}\nu_k & \varsigma_k^2 + \nu_k^2 - a \end{bmatrix}$$
$$= \begin{bmatrix} \ell_1 & & & \\ c_2 & \ell_2 & & \\ & \ddots & \ddots & \\ & & c_k & \ell_k \end{bmatrix} \begin{bmatrix} \ell_1 & c_2 & & \\ & \ell_2 & \ddots & \\ & & \ddots & c_k \\ & & & \ell_k \end{bmatrix}.$$

It is easy to verify that the Cholesky factors are given by the recurrence relations

$$\ell_1 = \sqrt{\varsigma_1^2 - a}, \quad c_j = \varsigma_{j-1}\nu_j/\ell_{j-1}, \quad \ell_j = \sqrt{\varsigma_j^2 + \nu_j^2 - a - c_j^2}, \quad j = 2, 3, \ldots$$

We may now compute δ_k by first solving

$$\begin{bmatrix} \ell_1 & & & \\ c_2 & \ell_2 & & \\ & \ddots & \ddots & \\ & & c_k & \ell_k \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ -\varsigma_k^2\nu_{k+1}^2 \end{bmatrix},$$

i.e., $\boldsymbol{\pi}_k = (\pi_1, \ldots, \pi_k) = -\varsigma_k^2\nu_{k+1}^2/\ell_k\, \mathbf{e}_k$, and next extracting the last component of the solution to

$$\begin{bmatrix} \ell_1 & c_2 & & \\ & \ell_2 & \ddots & \\ & & \ddots & c_k \\ & & & \ell_k \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ -\varsigma_k^2\nu_{k+1}^2/\ell_k \end{bmatrix},$$

i.e,

$$\delta_k := -\frac{\varsigma_k^2\nu_{k+1}^2}{\ell_k^2} = -\frac{\varsigma_k^2\nu_{k+1}^2}{\varsigma_k^2 + \nu_k^2 - a - c_k^2},$$

with the special case that $\nu_k = c_k = 0$ when $k = 1$.

There remains to accumulate the terms of the quadrature. Having chosen a Gauss-Radau approximation to (7.5), we are interested in computing $\mathbf{e}_1^{\mathsf{T}}\tilde{\mathbf{T}}_{k+1}^{-1}\mathbf{e}_1$ with $\tilde{\mathbf{T}}_{k+1}$ defined as in (7.8). It is easy to verify that the Cholesky factorization $\tilde{\mathbf{L}}_{k+1}\tilde{\mathbf{L}}_{k+1}^{\mathsf{T}}$ of $\tilde{\mathbf{T}}_{k+1}$ is given by

$$\tilde{\mathbf{T}}_{k+1} = \begin{bmatrix} \hat{\mathbf{T}}_k & \varsigma_k\nu_{k+1}\mathbf{e}_k \\ \varsigma_k\nu_{k+1}\mathbf{e}_k^{\mathsf{T}} & \omega_{k+1} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{U}}_k^{\mathsf{T}} & \\ \nu_{k+1}\mathbf{e}_k^{\mathsf{T}} & u_{k+1} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{U}}_k & \nu_{k+1}\mathbf{e}_k \\ & u_{k+1} \end{bmatrix} = \tilde{\mathbf{L}}_{k+1}\tilde{\mathbf{L}}_{k+1}^{\mathsf{T}},$$

where $u_{k+1} = \sqrt{\omega_{k+1} - \nu_{k+1}^2}$. Then, $\mathbf{e}_1^{\mathsf{T}}\tilde{\mathbf{T}}_{k+1}^{-1}\mathbf{e}_1 = \|\tilde{\mathbf{L}}_{k+1}^{-1}\mathbf{e}_1\|_2^2$ and this last squared norm is accumulated into the variable Ξ^2 using the procedure

$$\begin{aligned} \chi_1 &= 1/\varsigma_1, & \Xi^2 &\leftarrow \chi_1^2, \\ \chi_{j+1} &= -\nu_{j+1}\chi_j/u_{j+1}, & \Xi^2 &\leftarrow \Xi^2 + \chi_{j+1}^2, \quad j = 1, 2, \ldots \end{aligned}$$

Observe that by definition $\mathbf{z}_k = \gamma \hat{\mathbf{U}}_k^{-\mathsf{T}} \mathbf{e}_1$ and therefore the first k components of $\boldsymbol{\chi}_k = (\chi_1, \ldots, \chi_k)$ are precisely equal to $\gamma^{-1}\mathbf{z}_k$. Since the main loop of the algorithm already computes $\mathbf{z}_k$, we need only compute the last $\chi_{k+1} = -\nu_{k+1}\zeta_k/(\gamma u_{k+1})$. Since the γ in the denominator appears in both the components of $\boldsymbol{\chi}_k$ and in Ξ, we remove it from both places.

Finally, we obtain the realization of the Gauss-Radau convergence test described in Algorithm 7.1. This algorithm should be interleaved with one of Algorithm 6.1, 6.2, 6.3 or 6.4, which we indicate with comments in lines 1 and 9.

Algorithm 7.1 Gauss-Radau Convergence Test

Require: $\hat{\mathbf{U}}_k$, $d \in \mathbb{N}_0$, $\tau \in (0,1)$, $a \in (0,1)$
1: // *Generate* ς_1 *and* ζ_1
2: Set $\nu_1 := 0$, $c_1 := 0$, $\chi_1 := \zeta_1$, $\Xi^2 := \chi_1^2$, $k := 1$ and converged := **false**.
3: **while** $k < k_{\max}$ **do**
4: $\quad \ell_k^2 = \varsigma_k^2 + \nu_k^2 - a - c_k^2, \quad \delta_k = \varsigma_k^2 \nu_{k+1}^2 / \ell_k^2, \quad c_{k+1}^2 = \nu_{k+1}^2 / \ell_k^2$
5: $\quad \omega_{k+1} = a + \delta_k, \quad u_{k+1} = \sqrt{\omega_{k+1} - \nu_{k+1}^2}$
6: $\quad \chi_{k+1} = -\nu_{k+1}\zeta_k / u_{k+1}, \quad \Xi^2 = \sum_{j=1}^{k} \zeta_j^2 + \chi_{k+1}^2$
7: $\quad$ **if** $k \geq d$ **then**
8: $\quad\quad$ converged $= \left(\sum_{j=k-d+1}^{k} \zeta_j^2 \leq \tau \Xi^2 \right)$
9: $\quad$ // *Compute* ζ_{k+1}

Theorem 6.4 of Golub and Meurant (2010) ensures that Algorithm 7.1 computes an upper bound on the direct error. In typical situations, the major inconvenient of the Gauss-Radau approach is the need for an accurate estimate of the smallest eigenvalue of $\hat{\mathbf{T}}$, which can be very difficult in general. It is remarkable that in our case, the nature of $\hat{\mathbf{T}}$ guarantees that any value $0 < a < 1$ is such an estimate and produces an upper bound on the error.

The choice of the delay d is driven by the application and the values of $\mathbf{M}$ and $\mathbf{N}$. If $\mathbf{M}$ and $\mathbf{N}$ can be chosen such that the elliptic singular values of $\mathbf{A}$ become bounded in an interval independent of n and m, the delay parameter d can be quite small. We expect this to be the case in certain fluid flow problems such as Stokes and stabilized Stokes for then, $\mathbf{M}$ and $\mathbf{N}$ are spectrally equivalent to operators defining appropriate norms in the relevant function spaces.

8. Full-Space Methods

In this section we investigate the relation between the methods of §6 and two well-known Lanczos-based methods applied directly to the SQD system (1.1) with appropriate right-hand side: the conjugate gradient method and the minimum residual method.

8.1. **Full-Space Lanczos Process: I.** Upon pasting the relations (4.1) together, we have

(8.1a)
$$\begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & -\mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{k+1} & \\ & \mathbf{V}_k \end{bmatrix} = \begin{bmatrix} \mathbf{M} & \\ & \mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{k+1} & \\ & \mathbf{V}_k \end{bmatrix} \begin{bmatrix} \mathbf{I}_{k+1} & \mathbf{E}_k \\ \mathbf{E}_k^T & -\mathbf{I}_k \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \alpha_{k+1}\mathbf{N}\mathbf{v}_{k+1} \end{bmatrix} \mathbf{e}_{2k+1}^T$$

(8.1b)
$$\begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & -\mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{U}_k & \\ & \mathbf{V}_k \end{bmatrix} = \begin{bmatrix} \mathbf{M} & \\ & \mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{U}_k & \\ & \mathbf{V}_k \end{bmatrix} \begin{bmatrix} \mathbf{I}_k & \mathbf{B}_k \\ \mathbf{B}_k^T & -\mathbf{I}_k \end{bmatrix} + \begin{bmatrix} \beta_{k+1}\mathbf{M}\mathbf{u}_{k+1} \\ \mathbf{0} \end{bmatrix} \mathbf{e}_{2k}^T.$$

We claim that (8.1) describe a Lanczos process on the coefficient matrix of (1.1) using a metric defined by the block diagonal matrix $\mathrm{blkdiag}(\mathbf{M}, \mathbf{N})$ and that the matrix generated after k steps is itself SQD. Indeed, using the permutation matrix $\mathbf{P} := \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_{k+1} & \mathbf{e}_2 & \mathbf{e}_{k+2} & \cdots & \mathbf{e}_k & \mathbf{e}_{2k} \end{bmatrix}$, we have

$$\mathbf{P}^T \begin{bmatrix} \mathbf{I}_{k+1} & \mathbf{E}_k \\ \mathbf{E}_k^T & -\mathbf{I}_k \end{bmatrix} \mathbf{P} = \mathbf{T}_{2k+1} := \begin{bmatrix} 1 & \alpha_1 & & & & & \\ \alpha_1 & -1 & \beta_1 & & & & \\ & \beta_1 & 1 & \alpha_2 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \alpha_k & -1 & \beta_{k+1} \\ & & & & \beta_{k+1} & 1 \end{bmatrix} \tag{8.2}$$

$$= \begin{bmatrix} \mathbf{T}_{2k} & \beta_{k+1}\mathbf{e}_{2k} \\ \beta_{k+1}\mathbf{e}_{2k}^T & 1 \end{bmatrix}, \tag{8.3}$$

which is the tridiagonal matrix $\mathbf{T}_{2k+1}$ generated after $2k+1$ steps of the Lanczos process described above. The Lanczos vectors $\mathbf{s}_k$ generated by the above process have the form $\mathbf{s}_{2k+1} := (\mathbf{u}_k, \mathbf{0})$ and $\mathbf{s}_{2k+2} := (\mathbf{0}, \mathbf{v}_k)$ for $k \geq 0$. Moreover, the permutation $\mathbf{P}$ restores the order in which those vectors are generated by the algorithm, i.e.,

$$\mathbf{P}^T \begin{bmatrix} \mathbf{U}_k & \\ & \mathbf{V}_k \end{bmatrix} \mathbf{P} = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_{2k} \end{bmatrix}.$$

8.2. **Relation with the Direct Lanczos Method.** According to Definition 1.1, $\mathbf{T}_{2k}$ and $\mathbf{T}_{2k+1}$ defined in (8.2) and (8.3) are symmetric and quasi-definite. They therefore possess the Cholesky-like factorizations without pivoting:

$$\mathbf{T}_{2k} = \mathbf{L}_{2k}\mathbf{D}_{2k}\mathbf{L}_{2k}^T$$

and, using (8.2)–(8.3),

$$\begin{aligned} \mathbf{T}_{2k+1} &= \begin{bmatrix} \mathbf{L}_{2k}\mathbf{D}_{2k}\mathbf{L}_{2k}^T & \beta_{k+1}\mathbf{e}_{2k} \\ \beta_{k+1}\mathbf{e}_{2k}^T & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{L}_{2k} & \\ \ell_{2k}\mathbf{e}_{2k}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{D}_{2k} & \\ & d_{2k+1} \end{bmatrix} \begin{bmatrix} \mathbf{L}_{2k}^T & \ell_{2k}\mathbf{e}_{2k} \\ & 1 \end{bmatrix} \\ &= \mathbf{L}_{2k+1}\mathbf{D}_{2k+1}\mathbf{L}_{2k+1}^T, \end{aligned}$$

where $d_{2k+1} > 0$. Similarly, $\mathbf{T}_{2k+2}$ is given by

$$\begin{bmatrix} \mathbf{L}_{2k+1} & \\ \ell_{2k+1}\mathbf{e}_{2k+1}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{D}_{2k+1} & \\ & d_{2k+2} \end{bmatrix} \begin{bmatrix} \mathbf{L}_{2k+1}^T & \ell_{2k+1}\mathbf{e}_{2k+1} \\ & 1 \end{bmatrix} = \mathbf{L}_{2k+2}\mathbf{D}_{2k+2}\mathbf{L}_{2k+2}^T,$$

where $d_{2k+2} < 0$. The factors are computed using the recursions

$$d_1 = 1, \quad d_{j+1} = t_{j+1,j+1} - d_j \ell_j^2, \quad \ell_j = t_{j+1,j}/d_j, \quad j = 1, 2, \ldots$$

The entries $t_{i,j}$ of $\mathbf{T}_k$ are given by (8.2), i.e., $t_{j,j} = (-1)^{j+1}$, $t_{2j,2j-1} = \alpha_j$ and $t_{2j+1,2j} = \beta_j$.

The direct Lanczos method, referred to as DLANCZOS by Saad (2003, Algorithm 6.17), is simply the Lanczos process in which the tridiagonal system with coefficient $\mathbf{T}_k$ and with right-hand side $\beta_1 \mathbf{e}_1$ is solved at each iteration. Our approach is to compute the factors $\mathbf{L}_k$ and $\mathbf{D}_k$ of $\mathbf{T}_k$ and update them at each iteration. Systems involving $\mathbf{T}_k$ are then solved by way of the usual forward and backward substitutions. Note that each $\mathbf{L}_k$ is unit lower bidiagonal.

Let $k = 2i$ be an even iteration number. Consider the system (1.1) with right-hand side $(\mathbf{b}, \mathbf{0})$ and an approximation in the k-th Krylov subspace of the form $(\mathbf{x}_k, \mathbf{y}_k) = (\mathbf{U}_k \bar{\mathbf{x}}_k, \mathbf{V}_k \bar{\mathbf{y}}_k)$. Upon premultiplying (1.1) with $\operatorname{blkdiag}(\mathbf{U}_k^T, \mathbf{V}_k^T)$ and using (8.1b), the $\mathbf{M}$-orthogonality of the vectors $\{\mathbf{u}_k\}$ and the $\mathbf{N}$-orthogonality of the vectors $\{\mathbf{v}_k\}$, we obtain (6.13), which is a step of G-CRAIG and is precisely the subproblem solved at iteration k of DLANCZOS. Observe that the choice $\bar{\mathbf{y}}_k = \mathbf{B}_k^T \bar{\mathbf{x}}_k$ automatically satisfies the second block of the equations (6.13) when $\bar{\mathbf{x}}_k$ solves (6.17). Consider now an approximation in the $(k+1)$-st Krylov space of the form $(\mathbf{x}_{k+1}, \mathbf{y}_k) = (\mathbf{U}_{k+1} \bar{\mathbf{x}}_{k+1}, \mathbf{V}_k \bar{\mathbf{y}}_k)$. Proceeding as above, we obtain (6.3), which is a step of G-LSQR and is precisely the subproblem solved at iteration $k+1$ of DLANCZOS. In particular, $\bar{\mathbf{y}}_k$ satisfies the normal equations (6.1) and $\bar{\mathbf{x}}_{k+1}$ satisfies the first $(k+1)$ equations of (6.3).

This behaviour is a consequence of the expressions (5.16) and (5.17) for the Krylov spaces, which alternate between

$$\mathcal{K}_i\left(\bar{\mathbf{D}}, \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}\right) = \begin{bmatrix} \mathcal{K}_i(\bar{\mathbf{D}}_1, \mathbf{b}) \\ 0 \end{bmatrix} \quad \text{and} \quad \bar{\mathbf{K}}\mathcal{K}_i\left(\bar{\mathbf{D}}, \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}\right) = \begin{bmatrix} \mathcal{K}_i(\bar{\mathbf{D}}_1, \mathbf{b}) \\ \mathcal{K}_i(\bar{\mathbf{D}}_2, \bar{\mathbf{A}}^T \mathbf{b}) \end{bmatrix}.$$

The above shows that the DLANCZOS method does not break down when applied to the SQD system (1.1) with $\mathbf{f} = \mathbf{b}$ and $\mathbf{g} = \mathbf{0}$. Moreover, the underlying Lanczos process generates SQD matrices at each iteration. In particular, in exact arithmetic, the conjugate gradient method alternates between the minimization of the convex part of the quadratic form generated by the SQD matrix and the maximization of its concave part. In other words, upon denoting $\mathbf{e}_{\mathbf{x},k} = \mathbf{x} - \mathbf{x}_k$ and $\mathbf{e}_{\mathbf{y},k} = \mathbf{y} - \mathbf{y}_k$, the DLANCZOS method solves the problem

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}}\, \underset{\mathbf{y}}{\text{maximize}} \quad & \begin{bmatrix} \mathbf{e}_{\mathbf{x},k}^T & \mathbf{e}_{\mathbf{y},k}^T \end{bmatrix} \begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & -\mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{e}_{\mathbf{x},k} \\ \mathbf{e}_{\mathbf{y},k} \end{bmatrix} \\ & \text{subject to } (\mathbf{x}, \mathbf{y}) \in \mathcal{K}_k\left(\mathbf{H}^{-1}\mathbf{K}, \begin{bmatrix} \mathbf{M}^{-1}\mathbf{b} \\ 0 \end{bmatrix}\right). \end{aligned} \tag{8.4}$$

A consequence of the previous paragraphs and the stability analysis of the LDL^T factorization of SQD matrices due to Gill et al. (1996) is that there is no need for a symmetric indefinite factorization of $\mathbf{T}_k$ in the vein of Marcia (2008) for problem (1.1).

We have proved the following result.

Theorem 8.1. *The* DLanczos *method on the SQD system (1.1) with right-hand side* $(\mathbf{b}, \mathbf{0})$ *is well defined and will not break down. At each iteration, the tridiagonal matrix generated by the Lanczos process is SQD. Every odd step is a generalized* Lsqr *step. Every even step is a generalized* Craig *step.*

If we denote $\mathbf{z}_k = (\mathbf{x}_k, \mathbf{y}_k)$, $\mathbf{x}_k = \mathbf{U}_k \bar{\mathbf{x}}_k$, $\mathbf{y}_k = \mathbf{V}_k \bar{\mathbf{y}}_k$, and $\bar{\mathbf{z}}_k = (\bar{\mathbf{x}}_k, \bar{\mathbf{y}}_k)$, the tridiagonal system (6.13) may be written $\mathbf{T}_{2k} \bar{\mathbf{z}}_{2k} = \beta_1 \mathbf{e}_1$. Taking into account the factorization of $\mathbf{T}_k$, we obtain

$$\mathbf{z}_k = \mathbf{S}_k \bar{\mathbf{z}}_k = \mathbf{S}_k \mathbf{L}_k^{-\mathsf{T}} \mathbf{D}_k^{-1} \mathbf{L}_k^{-1} (\beta_1 \mathbf{e}_1) = \mathbf{W}_k \mathbf{q}_k,$$

where we defined

$$\mathbf{W}_k := \mathbf{S}_k \mathbf{L}_k^{-\mathsf{T}} \mathbf{D}_k^{-1} \qquad \text{and} \qquad \mathbf{q}_k := \mathbf{L}_k^{-1} (\beta_1 \mathbf{e}_1).$$

This is equivalent to the usual derivation of the DLanczos method based on the LU factorization of the Lanczos tridiagonal. However, our usage of the $\mathrm{LDL}^{\mathsf{T}}$ factorization highlights the fact that each $\mathbf{T}_k$ is SQD. The components $(\gamma_1, \ldots, \gamma_k)$ of $\mathbf{q}_k$ are easily found by recursion:

$$\gamma_1 = \beta_1, \qquad \gamma_{j+1} = -\ell_j \gamma_j, \quad j = 1, 2, \ldots$$

Similarly, using the equivalent identity $\mathbf{L}_k \mathbf{D}_k \mathbf{W}_k^{\mathsf{T}} = \mathbf{S}_k^{\mathsf{T}}$, we find the rows of $\mathbf{W}_k^{\mathsf{T}}$, i.e., vectors $\mathbf{w}_1, \ldots, \mathbf{w}_k$, recursively based on the vectors $\mathbf{s}_k$:

$$\mathbf{w}_1 = \mathbf{s}_1 / d_1, \qquad \mathbf{w}_{j+1} = (\mathbf{s}_{j+1} - \ell_j d_j \mathbf{w}_j) / d_{j+1}, \quad j = 1, 2, \ldots$$

Knowledge of the vectors $\mathbf{w}_k$ leads to an efficient update of $\mathbf{z}_k$, bypassing the computation of $\bar{\mathbf{z}}_k$ altogether:

$$\mathbf{z}_k = \mathbf{W}_k \mathbf{q}_k = \mathbf{z}_{k-1} + \gamma_k \mathbf{w}_k.$$

As we show below, the matrix $\mathbf{W}_k$ forms a partial factor of $\mathbf{K}$, as defined in (3.5). This is a form of orthonormality of the vectors $\mathbf{w}_k$ in spite of the fact that $\mathbf{K}$ is indefinite. Indeed, the $\mathrm{LDL}^{\mathsf{T}}$ factorization of $\mathbf{T}_k$ can be arranged so that $\mathbf{L}_k$ is lower bidiagonal, but not with unit diagonal and $\mathbf{D}_k$ has ± 1 on its diagonal. With this alternative factorization, we have the updates

$$\gamma_1 = \beta_1 / \ell_{1,1}, \qquad \gamma_{j+1} = -\ell_{j+1,j} \gamma_j / \ell_{j+1,j+1}, \quad j = 1, 2, \ldots \tag{8.5}$$

and

$$\mathbf{w}_1 = \mathbf{s}_1 / \ell_{1,1}, \qquad \mathbf{w}_{j+1} = (\mathbf{s}_{j+1} - \ell_{j+1,j} \mathbf{w}_j) / \ell_{j+1,j+1}, \quad j = 1, 2, \ldots \tag{8.6}$$

Thus without loss of generality, we may understand $\mathbf{D}_k$ as having diagonal elements $(-1)^{j+1}$.

Theorem 8.2. *Let $\mathbf{W}_k$ be defined as above and let $\mathbf{T}_k = \mathbf{L}_k\mathbf{D}_k\mathbf{L}_k^T$ where $\mathbf{L}_k$ is lower bidiagonal and $\mathbf{D}_k$ is diagonal. Then, for $k = 1, \ldots, n+m$, we have the partial factorization*

$$\mathbf{W}_k^T\mathbf{K}\mathbf{W}_k = \mathbf{D}_k^{-1}. \tag{8.7}$$

The DLANCZOS *iterates satisfy*

$$\mathbf{z}_k = \sum_{j=1}^{k} \gamma_j \mathbf{w}_j \tag{8.8}$$

and

$$\mathbf{z}_k^T\mathbf{K}\mathbf{z}_k = \sum_{j=1}^{k} \frac{\gamma_j^2}{d_j} = \sum_{j=1}^{\lceil k/2 \rceil} \frac{\gamma_{2j-1}^2}{d_{2j-1}} - \sum_{j=1}^{\lfloor k/2 \rfloor} \frac{\gamma_{2j}^2}{(-d_{2j})}, \tag{8.9}$$

as well as the error identity

$$(\mathbf{z} - \mathbf{z}_k)^T\mathbf{K}(\mathbf{z} - \mathbf{z}_k) = \sum_{j=k+1}^{n+m} \frac{\gamma_j^2}{d_j} = \sum_{j=k+1}^{\lceil \frac{n+m}{2} \rceil} \frac{\gamma_{2j-1}^2}{d_{2j-1}} - \sum_{j=k+1}^{\lfloor \frac{n+m}{2} \rfloor} \frac{\gamma_{2j}^2}{(-d_{2j})}, \tag{8.10}$$

where $\mathbf{z}$ is an exact solution of (1.1) with $\mathbf{f} = \mathbf{b}$ and $\mathbf{g} = \mathbf{0}$.

Proof. It suffices to note that, after applying the permutation $\mathbf{P}$, we have from (8.1b)

$$\mathbf{S}_k^T\mathbf{K}\mathbf{S}_k = \mathbf{S}_k^T\mathbf{H}\mathbf{S}_k\mathbf{T}_k = \mathbf{L}_k^T\mathbf{D}_k\mathbf{L}_k,$$

where $\mathbf{H}$ is defined in (3.6) and where we used the fact that the vectors $\mathbf{s}_k$ are orthonormal in the metric defined by $\mathbf{H}$. Introducing now the definition of $\mathbf{W}_k$, we obtain

$$\mathbf{W}_k^T\mathbf{K}\mathbf{W}_k = \mathbf{D}_k^{-1}\mathbf{L}_k^{-1}\mathbf{S}_k^T\mathbf{K}\mathbf{S}_k\mathbf{L}_k^{-T}\mathbf{D}_k^{-1} = \mathbf{D}_k^{-1}.$$

The rest of the proof follows directly, as in previous sections. □

It is important to note that $\mathbf{K}$, being indefinite, does not define a norm. It does however define a distance and this is why we do not use the norm notation in (8.9) and (8.10). Indeed, $d_{2k} < 0$ and $d_{2k+1} > 0$. Following the nomenclature of Gohberg, Lancaster, and Rodman (2005, Chapter 2), the vectors $\mathbf{w}_k$ are orthogonal in the metric $\mathbf{K}$ and orthonormal if the factorization of $\mathbf{T}_k$ is arranged so that

$$\mathbf{D}_{2k+1} = \mathbf{P}^T \begin{bmatrix} \mathbf{I}_{k+1} & \\ & -\mathbf{I}_k \end{bmatrix} \mathbf{P}.$$

For these reasons, we emphasized the negative terms in (8.9) and (8.10) by separating odd and even indices. The sum over odd indices corresponds to G-CRAIG iterations where $d_{2j-1} > 0$ and where the primal, minimum-norm, problem is solved. The sum over even indices corresponds to G-LSQR iterations where the dual, negative least-squares, problem is solved.

The conjugate gradient method is simply a reformulation of DLANCZOS in which the LU factorization of $\mathbf{T}_k$ is computed instead. Therefore, the conclusions above also apply to CG with appropriate redefinitions of $\mathbf{L}_k$ and $\mathbf{D}_k$. Interestingly, even in the present indefinite context, the conjugate gradient algorithm continues to perform its well-known minimization of the error in the energy "norm" with the

difference that it alternates between minimization steps in one problem and maximization steps in the dual problem viewed as a maximization problem. In particular, it is possible to define a corresponding stopping test based on the direct error in "energy norm". Select an integer $d > 0$ and a threshold $\tau > 0$. We may terminate the DLanczos iterations as soon as the partial sums (8.10) computed only over the past d iterations stabilize. More precisely, we terminate the iterations as soon as

$$\sum_{j=k+1}^{\lceil \frac{k+d+1}{2} \rceil} \frac{\gamma_{2j-1}^2}{d_{2j-1}} < \tau^2 \sum_{j=1}^{\lceil \frac{k+d+1}{2} \rceil} \frac{\gamma_{2j-1}^2}{d_{2j-1}} \quad \text{and} \quad \sum_{j=k+1}^{\lfloor \frac{k+d+1}{2} \rfloor} \frac{\gamma_{2j}^2}{(-d_{2j})} < \tau^2 \sum_{j=1}^{\lfloor \frac{k+d+1}{2} \rfloor} \frac{\gamma_{2j}^2}{(-d_{2j})}.$$

In exact arithmetic, it is equivalent to stop as soon as

$$\sum_{\substack{j=k+1 \\ d_j>0}}^{k+d+1} \frac{\gamma_j^2}{d_j} < \tau^2 \sum_{\substack{j=1 \\ d_j>0}}^{k+d+1} \frac{\gamma_j^2}{d_j} \quad \text{and} \quad \sum_{\substack{j=k+1 \\ d_j<0}}^{k+d+1} \frac{\gamma_j^2}{(-d_j)} < \tau^2 \sum_{\substack{j=1 \\ d_j<0}}^{k+d+1} \frac{\gamma_j^2}{(-d_j)}. \tag{8.11}$$

Alternatively, it is also possible to stop as soon as

$$\sum_{j=k+1}^{k+d+1} \frac{\gamma_j^2}{|d_j|} < \tau^2 \sum_{j=1}^{k+d+1} \frac{\gamma_j^2}{|d_j|}.$$

We stress that the error (8.10) measured in the metric $\mathbf{K}$ can be either positive or negative. Although we have not formally established this fact, in practice, its sign typically alternates, as does the sign of the pivots d_j, and exhibits an oscillatory behavior. It approaches zero in absolute value as k approaches $n + m$. This is illustrated in the numerical experiments of §9.

8.3. **Full-Space Lanczos Process: II.** The two methods g-LSMR and g-CRAIG-MR of §6.5 and §6.7 turn out to combine to become equivalent to an appropriately-preconditioned MINRES on (1.1). This result parallels Theorem 8.1 and the combination of g-LSQR and g-CRAIG to form g-CG. Upon pasting the Lanzcos processes (6.38) and (6.48) together, we obtain

$$\begin{bmatrix} \mathbf{M} + \mathbf{A}\mathbf{N}^{-1}\mathbf{A}^\mathsf{T} & \\ & \mathbf{N} + \mathbf{A}^\mathsf{T}\mathbf{M}^{-1}\mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{U}_k & \\ & \mathbf{V}_k \end{bmatrix} = \begin{bmatrix} \mathbf{M} & \\ & \mathbf{N} \end{bmatrix} \left(\begin{bmatrix} \mathbf{U}_k & \\ & \mathbf{V}_k \end{bmatrix} \begin{bmatrix} \mathbf{B}_k\mathbf{B}_k^\mathsf{T} + \mathbf{I}_k & \\ & \mathbf{E}_k^\mathsf{T}\mathbf{E}_k + \mathbf{I}_k \end{bmatrix} + \begin{bmatrix} \alpha_k\beta_{k+1}\mathbf{u}_{k+1}\mathbf{e}_k^\mathsf{T} & \\ & \alpha_{k+1}\beta_{k+1}\mathbf{v}_{k+1}\mathbf{e}_k^\mathsf{T} \end{bmatrix} \right). \tag{8.12}$$

As already noted in (5.10),

$$\begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^\mathsf{T} & -\mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{M}^{-1} & \\ & \mathbf{N}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^\mathsf{T} & -\mathbf{N} \end{bmatrix} = \begin{bmatrix} \mathbf{M} + \mathbf{A}\mathbf{N}^{-1}\mathbf{A}^\mathsf{T} & \\ & \mathbf{N} + \mathbf{A}^\mathsf{T}\mathbf{M}^{-1}\mathbf{A} \end{bmatrix}.$$

Upon premultiplying (8.12) by blkdiag($\mathbf{U}_k^\mathsf{T}, \mathbf{V}_k^\mathsf{T}$) and using the $\mathbf{M}$-orthogonality of the vectors $\mathbf{u}_k$ and the $\mathbf{N}$-orthogonality of the vectors $\mathbf{v}_k$, the Lanczos process may

be equivalently rewritten

$$\begin{bmatrix}\mathbf{U}_k & \\ & \mathbf{V}_k\end{bmatrix}^T \begin{bmatrix}\mathbf{M}+\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^T & \\ & \mathbf{N}+\mathbf{A}^T\mathbf{M}^{-1}\mathbf{A}\end{bmatrix} \begin{bmatrix}\mathbf{U}_k & \\ & \mathbf{V}_k\end{bmatrix} = \begin{bmatrix}\mathbf{B}_k\mathbf{B}_k^T+\mathbf{I}_k & \\ & \mathbf{E}_k^T\mathbf{E}_k+\mathbf{I}_k\end{bmatrix} = \begin{bmatrix}\mathbf{B}_k\mathbf{B}_k^T+\mathbf{I}_k & \\ & \mathbf{B}_k^T\mathbf{B}_k+\mathbf{I}_k\end{bmatrix} + \beta_{k+1}^2\mathbf{e}_{2k}\mathbf{e}_{2k}^T.$$

8.4. **Relation with the Minimum Residual Method.** In MINRES, the system (1.1) with $\mathbf{f} = \mathbf{b}$ and $\mathbf{g} = \mathbf{0}$ is tackled directly using sequential approximations in a Lanczos subspace by iteratively minimizing the norm of the residual. Typically, the Euclidian norm is used but other norms can be used via preconditioning. It turns out that the relevant Lanczos process in this case is precisely (8.1) and the k-th Krylov subspace $\mathcal{K}_k$ is spanned by the vectors $(\mathbf{u}_k, \mathbf{0})$ and $(\mathbf{0}, \mathbf{v}_k)$. Let $(\mathbf{x}_k, \mathbf{y}_k)$ be our approximation of $(\mathbf{x}, \mathbf{y})$ in $\mathcal{K}_k$. The corresponding residual is given by

$$\begin{bmatrix}\mathbf{r}_k \\ \mathbf{s}_k\end{bmatrix} := \begin{bmatrix}\mathbf{b} \\ \mathbf{0}\end{bmatrix} - \begin{bmatrix}\mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & -\mathbf{N}\end{bmatrix}\begin{bmatrix}\mathbf{x}_k \\ \mathbf{y}_k\end{bmatrix} \tag{8.13}$$

$$= \begin{bmatrix}\beta_1\mathbf{M}\mathbf{u}_1 \\ \mathbf{0}\end{bmatrix} - \begin{bmatrix}\mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & -\mathbf{N}\end{bmatrix}\begin{bmatrix}\mathbf{U}_k & \\ & \mathbf{V}_k\end{bmatrix}\begin{bmatrix}\bar{\mathbf{x}}_k \\ \bar{\mathbf{y}}_k\end{bmatrix}, \tag{8.14}$$

for some vector $(\bar{\mathbf{x}}_k, \bar{\mathbf{y}}_k)$. In this section we show that the Lanczos process (8.1) determines an implementation of MINRES in which the residual (8.13) is minimized in the norm defined by $\text{blkdiag}(\mathbf{M}^{-1}, \mathbf{N}^{-1})$, which imposes that $(\bar{\mathbf{x}}_k, \bar{\mathbf{y}}_k)$ satisfy the Ritz-Galerkin condition

$$\begin{bmatrix}\mathbf{U}_k & \\ & \mathbf{V}_k\end{bmatrix}^T \begin{bmatrix}\mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & -\mathbf{N}\end{bmatrix}\begin{bmatrix}\mathbf{M}^{-1} & \\ & \mathbf{N}^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{r}_k \\ \mathbf{s}_k\end{bmatrix} = \begin{bmatrix}\mathbf{0} \\ \mathbf{0}\end{bmatrix}. \tag{8.15}$$

Upon premultiplying (8.1b) with

$$\begin{bmatrix}\mathbf{U}_k & \\ & \mathbf{V}_k\end{bmatrix}^T \begin{bmatrix}\mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & -\mathbf{N}\end{bmatrix}^T \begin{bmatrix}\mathbf{M}^{-1} & \\ & \mathbf{N}^{-1}\end{bmatrix},$$

we obtain a sum of two terms. The first term is

$$\begin{bmatrix}\mathbf{U}_k^T & \\ & \mathbf{V}_k^T\end{bmatrix}\begin{bmatrix}\mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & -\mathbf{N}\end{bmatrix}\begin{bmatrix}\mathbf{U}_k & \\ & \mathbf{V}_k\end{bmatrix}\begin{bmatrix}\mathbf{I}_k & \mathbf{B}_k \\ \mathbf{B}_k^T & -\mathbf{I}_k\end{bmatrix}.$$

Using (8.1b) itself, this term may be rewritten

$$\begin{bmatrix}\mathbf{U}_k^T & \\ & \mathbf{V}_k^T\end{bmatrix}\begin{bmatrix}\mathbf{M} & \\ & \mathbf{N}\end{bmatrix}\begin{bmatrix}\mathbf{U}_k & \\ & \mathbf{V}_k\end{bmatrix}\begin{bmatrix}\mathbf{I}_k & \mathbf{B}_k \\ \mathbf{B}_k^T & -\mathbf{I}_k\end{bmatrix}^2 + \begin{bmatrix}\mathbf{U}_k^T & \\ & \mathbf{V}_k^T\end{bmatrix}\begin{bmatrix}\beta_{k+1}\mathbf{M}\mathbf{u}_{k+1} \\ \mathbf{0}\end{bmatrix}\mathbf{e}_{2k}^T\begin{bmatrix}\mathbf{I}_k & \mathbf{B}_k \\ \mathbf{B}_k^T & -\mathbf{I}_k\end{bmatrix} = \begin{bmatrix}\mathbf{I}_k & \mathbf{B}_k \\ \mathbf{B}_k^T & -\mathbf{I}_k\end{bmatrix}^2,$$

where we used the $\mathbf{M}$-orthogonality of $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ and the $\mathbf{N}$-orthogonality of $\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$. The second term is

$$\begin{bmatrix} \mathbf{U}_k & \\ & \mathbf{V}_k \end{bmatrix}^T \begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & -\mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{M}^{-1} & \\ & \mathbf{N}^{-1} \end{bmatrix} \begin{bmatrix} \beta_{k+1}\mathbf{M}\mathbf{u}_{k+1} \\ \mathbf{0} \end{bmatrix} \mathbf{e}_{2k}^T = \beta_{k+1}^2 \mathbf{e}_{2k}\mathbf{e}_{2k}^T,$$

where we used (8.1b) and the orthogonality properties of the Lanczos vectors one more time, in the same way as for the first term. We have just showed that

$$\begin{bmatrix} \mathbf{U}_k & \\ & \mathbf{V}_k \end{bmatrix}^T \begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & -\mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{M}^{-1} & \\ & \mathbf{N}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & -\mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{U}_k & \\ & \mathbf{V}_k \end{bmatrix} = \begin{bmatrix} \mathbf{I}_k & \mathbf{B}_k \\ \mathbf{B}_k^T & -\mathbf{I}_k \end{bmatrix}^2 + \beta_{k+1}^2 \mathbf{e}_{2k}\mathbf{e}_{2k}^T \tag{8.16}$$

is the underlying Lanczos process governing the minimum residual method. The first matrix in the right-hand side of this last equality is the square of a symmetric permutation of $\mathbf{T}_{2k}$. It is thus a symmetric permutation of the pentadiagonal matrix $\mathbf{T}_{2k}^2$.

On the other hand, since

$$\begin{bmatrix} \mathbf{U}_k & \\ & \mathbf{V}_k \end{bmatrix}^T \begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & -\mathbf{N} \end{bmatrix}^T \begin{bmatrix} \mathbf{M}^{-1} & \\ & \mathbf{N}^{-1} \end{bmatrix} \begin{bmatrix} \beta_1 \mathbf{M}\mathbf{u}_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \beta_1 \mathbf{e}_1 \\ \alpha_1\beta_1 \mathbf{e}_1 \end{bmatrix},$$

the condition (8.15) amounts to the (psychologically) pentadiagonal system

$$\left(\begin{bmatrix} \mathbf{I}_k & \mathbf{B}_k \\ \mathbf{B}_k^T & -\mathbf{I}_k \end{bmatrix}^2 + \beta_{k+1}^2 \mathbf{e}_{2k}\mathbf{e}_{2k}^T \right) \begin{bmatrix} \bar{\mathbf{x}}_k \\ \bar{\mathbf{y}}_k \end{bmatrix} = \begin{bmatrix} \beta_1 \mathbf{e}_1 \\ \alpha_1\beta_1 \mathbf{e}_1 \end{bmatrix}. \tag{8.17}$$

The system (8.17) is precisely the system solved by the standard MINRES at iteration k (Paige and Saunders, 1975, Equations (6.1) and (6.2)). It is also easy to verify that (8.17) represents the optimality conditions of the linear least-squares problem

$$\underset{\bar{\mathbf{x}},\bar{\mathbf{y}}}{\text{minimize}} \ \tfrac{1}{2} \left\| \begin{bmatrix} \mathbf{I}_{k+1} & \mathbf{E}_k \\ \mathbf{E}_k^T & -\mathbf{I}_k \end{bmatrix} \begin{bmatrix} \bar{\mathbf{x}} \\ \bar{\mathbf{y}} \end{bmatrix} - \begin{bmatrix} \beta_1 \mathbf{e}_1 \\ \mathbf{0} \end{bmatrix} \right\|_2^2.$$

As a consequence, we obtain the generalized MINRES by substituting the standard Lanczos process used at each iteration of MINRES with (8.1). We have just proved the following theorem.

Theorem 8.3. *The generalized* MINRES *iterates on (1.1) with right-hand side* $(\mathbf{b}, \mathbf{0})$ *are the same as those generated by* MINRES *on (1.1) with preconditioner* $\mathit{blkdiag}(\mathbf{M}^{-1}, \mathbf{N}^{-1})$. *In addition, every even step is a* G-LSMR *step and every odd step is a* G-CRAIG-MR *step.*

Proof. It suffices to note that

$$\begin{bmatrix} \mathbf{B}_k\mathbf{B}_k^T + \mathbf{I}_k & \\ & \mathbf{B}_k^T\mathbf{B}_k + \mathbf{I}_k \end{bmatrix} = \begin{bmatrix} \mathbf{I}_k & \mathbf{B}_k \\ \mathbf{B}_k^T & -\mathbf{I}_k \end{bmatrix}^2$$

and that the Lanczos processes (8.12) and (8.16) are identical. This common Lanczos process implies that the G-MINRES method applied to (1.1) with right-hand

side $(\mathbf{b}, \mathbf{0})$ and preconditioner blkdiag$(\mathbf{M}^{-1}, \mathbf{N}^{-1})$ alternates between G-CRAIG-MR steps and G-LSMR steps. □

By definition of MINRES, the quantity $\|\mathbf{M}^{-\frac{1}{2}}\mathbf{r}_k\|_2^2 + \|\mathbf{N}^{-\frac{1}{2}}\mathbf{s}_k\|_2^2 = \|\mathbf{r}_k\|_{\mathbf{M}^{-1}}^2 + \|\mathbf{s}_k\|_{\mathbf{N}^{-1}}^2$ is nonincreasing with k. As we now show, MINRES also lends itself to an interpretation in terms of direct error, as opposed to residual, and a related stopping condition emerges. We begin with generalities on MINRES that lead to a new result indicating the appropriate norm in which MINRES measures direct errors. Because this result is general and always applies to MINRES, it is given in a separate section. We next specialize this result to the current SQD framework to provide an interpretation of MINRES as a decoupled combination of G-LSMR and G-CRAIG-MR.

8.4.1. *Generalities on* MINRES. In this section, we use the same notation as in §3. Consider a generic linear system $\mathbf{Hx} = \mathbf{d}$ where $\mathbf{H} = \mathbf{H}^\mathsf{T}$. MINRES generates Lanczos vectors $\mathbf{s}_k$ and a symmetric tridiagonal matrix $\mathbf{\Omega}_k$ according to (3.1). The process is summarized by (3.2). Much as in the previous section, MINRES can be summarized with the identity

$$\mathbf{S}_k^\mathsf{T}\mathbf{H}^2\mathbf{S}_k = \mathbf{\Omega}_k^2 + \beta_{k+1}^2\mathbf{e}_k\mathbf{e}_k^\mathsf{T}, \tag{8.18}$$

where the columns of $\mathbf{S}_k$ are theoretically orthonormal. Paige and Saunders (1975) compute the LQ factorization of $\mathbf{\Omega}_k$ and show that

$$\mathbf{\Omega}_k^2 + \beta_{k+1}^2\mathbf{e}_k\mathbf{e}_k^\mathsf{T} = \mathbf{L}_k\mathbf{L}_k^\mathsf{T}, \tag{8.19}$$

where $\mathbf{L}_k$ is lower tridiagonal. The iterates are updated according to

$$\mathbf{x}_k = \mathbf{W}_k\mathbf{t}_k = \sum_{j=1}^{k} \tau_j\mathbf{w}_j = \mathbf{x}_{k-1} + \tau_k\mathbf{w}_k,$$

where $\mathbf{t}_k = (\tau_1, \ldots, \tau_k)$ and $\mathbf{W}_k := \mathbf{S}_k\mathbf{L}_k^{-\mathsf{T}}$. The scalar τ_k is easily obtained by way of a recurrence at each iteration. We refer the reader to (Paige and Saunders, 1975, Section 6) for details. The above is sufficient to establish the following result.

Theorem 8.4. *Let* $\mathbf{W}_k$ *be defined as above. Then, for* $k = 1, \ldots, n$*, we have the partial factorization*

$$\mathbf{W}_k^{\mathsf{T}} \mathbf{H}^2 \mathbf{W}_k = \mathbf{I}_k. \tag{8.20}$$

The MINRES *iterates satisfy*

$$\mathbf{x}_k = \sum_{j=1}^{k} \tau_j \mathbf{w}_j, \tag{8.21}$$

and

$$\|\mathbf{x}_k\|_{\mathbf{H}^2}^2 = \sum_{j=1}^{k} \tau_j^2, \tag{8.22}$$

as well as the error identity

$$\|\mathbf{x} - \mathbf{x}_k\|_{\mathbf{H}^2}^2 = \sum_{j=k+1}^{n} \tau_j^2, \tag{8.23}$$

where $\mathbf{x}$ *is the solution of* $\mathbf{H}\mathbf{x} = \mathbf{d}$.

Proof. It suffices to note that

$$\mathbf{W}_k^{\mathsf{T}} \mathbf{H}^2 \mathbf{W}_k = \mathbf{L}_k^{-1} \mathbf{S}_k^{\mathsf{T}} \mathbf{H}^2 \mathbf{S}_k \mathbf{L}_k^{-\mathsf{T}} = \mathbf{L}_k^{-1} \mathbf{L}_k \mathbf{L}_k^{\mathsf{T}} \mathbf{L}_k^{-\mathsf{T}} = \mathbf{I}_k,$$

where we used (8.18) and (8.19). The rest of the proof is analogous to those of previous similar results. □

In the next section, we specialize Theorem 8.4 to the SQD context.

8.4.2. *An Error Estimate for* MINRES. In the SQD context, the generic identity (8.18) is paralleled by (8.16). The matrix multiplied left and right by the Lanczos vectors in the left-hand side of (8.16) is precisely the block-diagonal matrix $\mathbf{D}$ given by (5.9). MINRES performs an LQ factorization of

$$\begin{bmatrix} \mathbf{I}_k & \mathbf{B}_k \\ \mathbf{B}_k^{\mathsf{T}} & -\mathbf{I}_k \end{bmatrix}$$

in such a way that

$$\begin{bmatrix} \mathbf{U}_k & \\ & \mathbf{V}_k \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \mathbf{M} + \mathbf{A}\mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}} & \\ & \mathbf{N} + \mathbf{A}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{U}_k & \\ & \mathbf{V}_k \end{bmatrix} = \mathbf{L}_k \mathbf{L}_k^{\mathsf{T}},$$

where $\mathbf{L}_k$ is lower tridiagonal. The following result is a simple rewrite of Theorem 8.4.

Corollary 8.5. *Let $\mathbf{W}_k$ be defined as in Theorem 8.4. Then, for $k = 1, \ldots, n+m$, we have the partial factorization*

$$\mathbf{W}_k^T \mathbf{D} \mathbf{W}_k = \mathbf{I}_k \tag{8.24}$$

where $\mathbf{D}$ is defined in (5.9). In addition, the MINRES *iterates satisfy*

$$\begin{bmatrix} \mathbf{x}_k \\ \mathbf{y}_k \end{bmatrix} = \sum_{j=1}^{k} \tau_j \mathbf{w}_j, \tag{8.25}$$

and

$$\|(\mathbf{x}_k, \mathbf{y}_k)\|_{\mathbf{D}}^2 = \|\mathbf{x}_k\|_{\mathbf{M}+\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^T}^2 + \|\mathbf{y}_k\|_{\mathbf{N}+\mathbf{A}^T\mathbf{M}^{-1}\mathbf{A}}^2 = \sum_{j=1}^{k} \tau_j^2, \tag{8.26}$$

as well as the error identity

$$\begin{aligned} \|(\mathbf{x} - \mathbf{x}_k, \mathbf{y} - \mathbf{y}_k)\|_{\mathbf{D}}^2 &= \|\mathbf{x} - \mathbf{x}_k\|_{\mathbf{M}+\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^T}^2 + \|\mathbf{y} - \mathbf{y}_k\|_{\mathbf{N}+\mathbf{A}^T\mathbf{M}^{-1}\mathbf{A}}^2 \\ &= \sum_{j=k+1}^{n+m} \tau_j^2, \end{aligned} \tag{8.27}$$

where $(\mathbf{x}, \mathbf{y})$ is the solution of (1.1) with $\mathbf{f} = \mathbf{b}$ and $\mathbf{g} = \mathbf{0}$.

In particular, applying MINRES on (1.1) consists in applying G-LSMR in the variable $\mathbf{y}$ and G-CRAIG-MR in the variable $\mathbf{x}$ in a decoupled manner. This comes from the fact that $\mathbf{D}$ is block diagonal. It is then clear that, by contrast with CG, MINRES performs twice the work as it applies both G-LSMR and G-CRAIG-MR and terminates when both have reached satisfactory accuracy.

Note that the energy norm given in (8.26) and (8.27) differs from that of Silvester and Simoncini (2011), who use the norm defined by the matrix $\mathbf{H}$ of (3.6) in the context of systems of partial differential equations. The two error norms are related in some cases such as the simulation of Stokes flows by way of a mixed finite-element discretization. In this case, $\mathbf{N}$ is typically assumed to spectrally equivalent to the appropriate mass matrix $\mathbf{Q}$ in the sense that there exist positive constants γ_1 and γ_2 such that

$$\gamma_1 \leq \frac{\mathbf{q}^T(\mathbf{N} + \mathbf{A}^T\mathbf{M}^{-1}\mathbf{A})\mathbf{q}}{\mathbf{q}^T\mathbf{Q}\mathbf{q}} \leq \gamma_2$$

for all appropriate vectors $\mathbf{q}$ (Elman et al., 2005). Similarly, it is typically assumed that

$$\gamma_1 \leq \frac{\mathbf{v}^T(\mathbf{M} + \mathbf{A}\mathbf{Q}^{-1}\mathbf{A}^T)\mathbf{v}}{\mathbf{v}^T\mathbf{M}\mathbf{v}} \leq \gamma_2$$

for all appropriate vectors $\mathbf{v}$. Under such assumptions, the preconditioner $\mathbf{H}$ defines a natural error norm and it is spectrally equivalent to the error norm of Corollary 8.5.

9. Implementation and Numerical Experiments

Implementing the methods of Section 6 can be done by modifying existing implementations of the standard LSQR and LSMR. Each code has two lines in its initialization that compute the first Golub-Kahan vectors $\bar{u}_1$ and $\bar{v}_1$. Those two lines should be replaced by those of Algorithm 4.2 that compute u_1 and v_1. Similarly, the two lines in the main loop that compute $\bar{u}_{j+1}$ and $\bar{v}_{j+1}$ should be replaced

by the two lines in the main loop of Algorithm 4.2 using appropriate callbacks to solve systems with $\mathbf{M}$ and $\mathbf{N}$. Our implementation is in the Python language using the LSQR and LSMR implementations from the PyKrylov package (Orban, 2011). The implementations of CRAIG and CRAIG-MR are original.

We present below numerical results on two categories of test cases: problems originating from optimization and from discretized partial-differential equations for fluid flow. In both cases the stopping conditions for all methods are those described in Algorithms 6.1, 6.2, 6.3 and 6.4 with $\tau = 10^{-12}$. This results in more iterations than would typically be necessary but illustrates the decrease of the relevant error estimates to tight tolerances. The stopping test used for DLANCZOS is that described in (8.11). The latter proved more reliable than a test based on even and odd iterations as in practice, due to occasional instabilities in the factorization of the Lanczos tridiagonal, pivots do not always alternate sign at each iteration. As we explain below, it was necessary to select looser values of τ for this method. The stopping test used for MINRES is similar to those of §6 based on (8.27) with $\tau = 10^{-12}$.

Among other stopping conditions, both G-LSQR and G-LSMR declare optimality when either the relative residual or the relative residual of the normal equations falls below a certain threshold. More precisely, with $\epsilon_a = \epsilon_r = 10^{-12}$, the first stopping conditions is

$$\frac{\bar{r}_k}{\|\mathbf{b}\|_2} \leq \epsilon_a + \epsilon_r \frac{\|\mathbf{A}\|_2 \|\bar{\mathbf{y}}_k\|_2}{\|\mathbf{b}\|_2}, \tag{9.1}$$

where $\bar{r}_k$ is the value of the objective function of (2.4) at $\mathbf{y}_k = \mathbf{V}_k \bar{\mathbf{y}}_k$ and $\|\mathbf{A}\|_2$ is estimated by both G-LSQR and G-LSMR at each iteration. The second condition is

$$\frac{\bar{\rho}_k}{\|\mathbf{A}\|_2 \bar{r}_k} \leq \epsilon_a, \tag{9.2}$$

where $\bar{\rho}_k$ is the right-hand side of (6.37). In our experiments, those stopping conditions were never the reason for terminating.

For the purpose of the numerical illustration below, linear systems with coefficient $\mathbf{M}$ or $\mathbf{N}$ are solved using a one-time Cholesky factorization. In practice, the application should dictate the most appropriate solution method. It should be noted that no attempt was made to ensure clustering of the generalized singular values of $\mathbf{A}$. Each problem name is followed by a tuple (n, m) where n is the order of $\mathbf{M}$ and m is the order of $\mathbf{N}$. For each iterative method, we report the history of the *relative* direct error in the appropriate metric together with *relative* error estimates for $d = 5$ and $d = 15$. At each iteration k, the direct error estimates are the quantities of the general form (6.11) while relative error estimates have the general form

$$\left(\sum_{j=k-d+1}^{k} \zeta_j^2 / \sum_{j=1}^{k} \zeta_j^2 \right)^{\frac{1}{2}}.$$

The error metric for each method is as given by Theorems 6.2, 6.4, 6.6, 6.8, 8.2 and Corollary 8.5. Because DLANCZOS and MINRES combine two methods for the normal or Schur-complement equations, the value of d is doubled for them, i.e., setting $d = 5$ corresponds to an effective $d = 10$, so both underlying methods have time to converge.

Table 2 collects statistics on our test problems.

TABLE 2. Summary of test problems.

Name	Type	n	m	nnz($\mathbf{M}$)	nnz($\mathbf{A}$)	nnz($\mathbf{N}$)
DUAL1	Optimization	255	171	3728	425	171
STCQP1	Optimization	12291	10246	34797	29726	10246
COLLIDE	Stokes	578	289	2202	3465	1345
LID	Stokes	578	289	2202	3465	1345

9.1. **Problems from Optimization.** The two problems below are generated at the third iteration of the primal-dual regularized interior-point method of Friedlander and Orban (2012). They originate from quadratic programming problems in standard form

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}}\ \mathbf{g}^T\mathbf{x} + \tfrac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} \quad \text{subject to } \mathbf{C}\mathbf{x} = \mathbf{d},\ \mathbf{x} \geq 0,$$

where $\mathbf{g} \in \mathbb{R}^n$ and $\mathbf{H} = \mathbf{H}^T \in \mathbb{R}^{n\times n}$ is positive semi-definite, and result in linear systems with coefficient matrix

$$\begin{bmatrix} \mathbf{H} + \mathbf{X}^{-1}\mathbf{Z} + \rho I & \mathbf{C}^T \\ \mathbf{C} & -\delta\mathbf{I} \end{bmatrix}$$

where $\rho > 0$ and $\delta > 0$ are regularization parameters. The system is initially shifted as in §6.1 to recover a right-hand side with all zeros in its last m components. The quadratic programs are part of the CUTEr collection (Gould et al., 2003) and were chosen because they are representative of the behavior of the error curves. The numerical behavior of each method is illustrated in Figs 9.1 and 9.2.

Note that for both values of the window size d, the error estimates for G-LSQR and G-LSMR qualitatively follow the exact error although they underestimate it by one or two orders of magnitude. This is typical of the problems we tested In G-CRAIG and G-CRAIG-MR, the error estimates are not monotonic and this behavior is more apparent on DUAL1. The exact error curve for both methods exhibits a temporary plateau and at this point the error estimates try to recover from an under-estimation. Due to the window size, this recovery takes a number of iterations. The estimates otherwise closely follow the exact error curve. The hump in the error estimates is echoed in the G-MINRES curves, which combines G-LSMR and G-CRAIG-MR, located directly above it in the figure. Note that the hump occurs around iteration 80 for G-CRAIG-MR and around iteration 160 for MINRES, which consolidates the fact that every other MINRES step is a G-CRAIG-MR step.

The curves for DLANCZOS are less intuitive. Note first that Theorem 8.2 states that the error is measured in an indefinite metric while Theorem 8.1 explains that DLANCZOS steps alternate between G-LSQR and G-CRAIG steps, again located directly above the DLANCZOS plot in the figure. Following (8.10), the error changes between positive and negative values but globally decreases in absolute value. Figure 9.1 and those that follow plot the exact error curve for DLANCZOS on a *symmetric logarithmic scale*, i.e., a logarithmic scale in both positive and negative values. DLANCZOS turns out to be significantly less stable numerically than the other methods. Setting $\tau = 10^{-12}$ results in failure for several problems due to roundoff errors, possibly including fatal loss of orthogonality and instability of the factorization of the Lanczos tridiagonal. Similarly, a window size of 15 iterations resulted in failures for at least one problem. Those failures are consistently due to the norm of the

preconditioned residual becoming negative. Following this, DLANCZOS exits with an error message stating that the preconditioner is not positive definite. Another sign of numerical instability is that consecutive pivots occasionally have the same sign. We expect that an alternative implementation such as SYMMLQ (Paige and Saunders, 1975) would be more stable, although it is not yet clear how to recover the error estimates in SYMMLQ.

For the reasons above, the plots presented in this section for DLANCZOS use $\tau = 10^{-6}$ and window sizes of 5 and 10. In exact arithmetic we expect that the oscillations of the exact error should be contained between two enveloping monotonic curves. It is almost the case for STCQP1 but we see that the curve exhibits a spike for DUAL1. It is not entirely clear what the origin of this spike is but we speculate that it is partly due to DLANCZOS not exactly reducing to a decoupled combination of G-LSQR and G-CRAIG in finite-precision arithmetic. It also appears to overlap the plateau in the G-CRAIG error. The DLANCZOS error remains positive for a few iterations during which the G-CRAIG error does not decrease significantly. During those iterations, G-LSQR is essentially working alone towards reducing the error. For DUAL1, the error estimates are not particularly close to the exact error. For STCQP1, the error estimates follow the exact error more faithfully.

9.2. **Problems from Discretized PDEs.** The two test cases of this section are discretizations of the stabilized Stokes equations for incompressible fluid flow (1.2) over a two-dimensional domain. The mesh and discretization are generated by the software IFISS 3.1 of Elman et al. (2007). The problems originate from (Elman et al., 2005). In both problems, the domain is $\Omega := (-1, 1) \times (-1, 1)$ and the stabilization parameter is set to $\beta = 0.25$. These are examples where the regularization term $\mathbf{N}$ is not diagonal—in both examples, $\mathbf{N}$ is tridiagonal with semi-bandwidth 18. For both problems, the discretization occurs on a 16×16 grid with Q1-Q1 elements.

The first problem describes a colliding flow with analytic solution

$$u(x, y) = (20xy^3, 5x^4 - 5y^4), \quad p = 60x^2y - 20y^3 + \text{constant}$$

in Ω. Dirichlet boundary conditions are imposed along the whole boundary using the interpolant of the finite-element discretization of $u(x, y)$. Results are summarized in Fig. 9.3.

The second problem describes a flow in a lid-driven regularized cavity. The lid velocity is given by $u(x, y) = (1 - x^4, 0)$. Results are summarized in Figs. 9.3 and 9.4.

The behavior of the error estimates is smoother than in the case of optimization problems. In all cases the error estimates for $d = 5$ and $d = 15$ are superposed and follow closely the exact error curve. For G-LSQR, G-CRAIG, G-LSMR and G-CRAIG-MR, it would take of the order of 30 iterations to reduce the error by a factor of 10^6.

We set the DLANCZOS stopping tolerance to $\tau = 10^{-3}$, which is reasonable given the discretization step size. For tighter tolerances, DLANCZOS again fails, complaining about an indefinite preconditioner, which indicates that the method has been driven past the numerical convergence point.

In the case of the lid-driven cavity problem, the exact error curve for MINRES increases in the last few iterations and we believe that this is due to cancellation

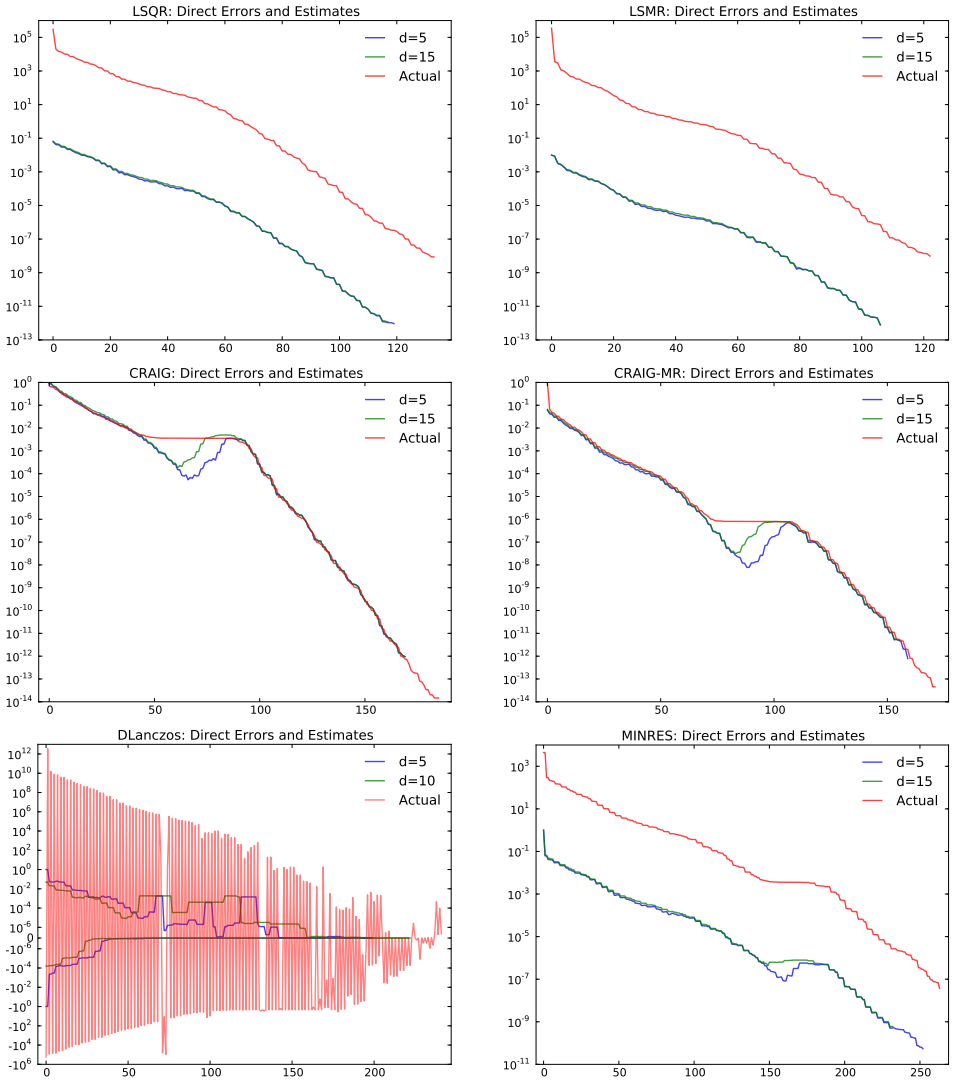

FIGURE 9.1. Problem DUAL1 (255, 171). Note the symmetric logarithmic scale of the vertical axis for DLANCZOS used to capture the fact that the error is measured in an indefinite metric.

when computing the error. This behavior is visible to a lesser extent in G-LSQR and G-LSMR.

10. DISCUSSION

In all instances, methods for the normal or Schur-complement equations are attractive because the lower bounds estimates of the direct error follow the same trend as the exact error. Our experiments illustrate that the two are tighter in Schur-complement equations methods. The behavior of DLANCZOS, and therefore or the conjugate gradient method and of SYMMLQ, on SQD systems is instructive.

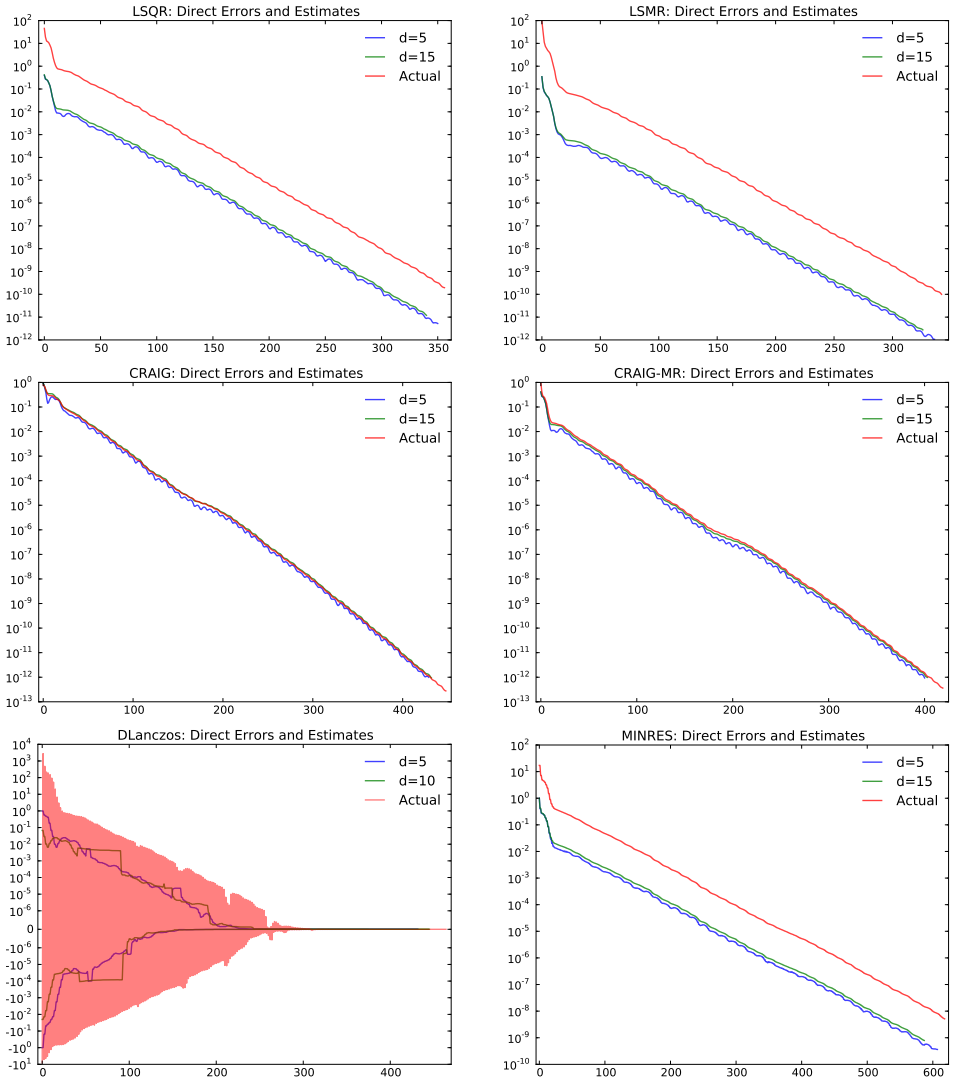

FIGURE 9.2. Problem STCQP1 (12291, 10246). Note the symmetric logarithmic scale of the vertical axis for DLANCZOS used to capture the fact that the error is measured in an indefinite metric.

Those methods solve the min-max problem

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}}\,\underset{\mathbf{y}\in\mathbb{R}^m}{\text{maximize}}\;\; \mathcal{L}(\mathbf{x},\mathbf{y})$$

where

$$\mathcal{L}(\mathbf{x},\mathbf{y}) := \tfrac{1}{2}\|\mathbf{x}\|_2^2 + \mathbf{x}^{\mathsf{T}}\bar{\mathbf{A}}\mathbf{y} - \tfrac{1}{2}\|\mathbf{y}\|_2^2 - \mathbf{f}^{\mathsf{T}}\mathbf{M}^{-\frac{1}{2}}\mathbf{x} - \mathbf{g}^{\mathsf{T}}\mathbf{N}^{-\frac{1}{2}}\mathbf{y},$$

whose first-order optimality conditions coincide with (1.1) preconditioned with $\mathbf{H}$ defined in (3.6), and in which the sign of the error alternates. Every other step is a minimization step on the convex part of $\mathcal{L}$, i.e., the function $\mathbf{x} \mapsto \mathcal{L}(\mathbf{x},\mathbf{y})$ for

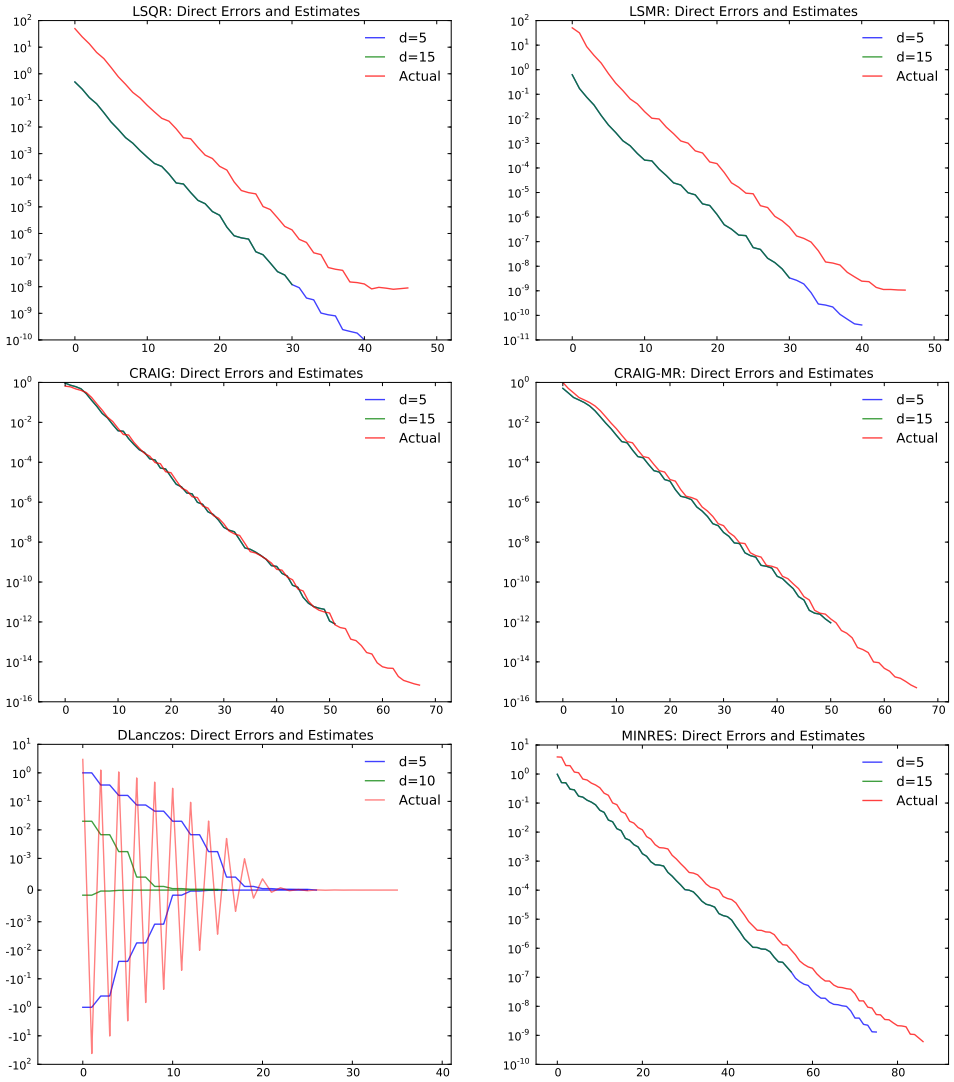

FIGURE 9.3. Colliding Flow $(578, 289)$. Note the symmetric logarithmic scale of the vertical axis for DLANCZOS used to capture the fact that the error is measured in an indefinite metric.

fixed $\mathbf{y}$, while the next step is a maximization step on the concave part of $\mathcal{L}$, i.e., $\mathbf{y} \mapsto \mathcal{L}(\mathbf{x}, \mathbf{y})$ for fixed $\mathbf{x}$.

It is possible to develop an upper bound estimate of the direct error for full-space methods using the same principles as in §7. In full-space methods, the tridiagonal $\hat{\mathbf{T}}$ has the form (8.2) and is itself SQD. Implementing a Gauss-Radau upper bound requires an accurate estimate of the smallest eigenvalue of $\hat{\mathbf{T}}$. Theorem 5.1 indicates that this smallest eigenvalue is $-\sqrt{1 + \sigma_{\max}^2}$ where $\sigma_{\max}$ is the largest elliptic singular value of $\mathbf{A}$. In general, such an estimate is not directly available.

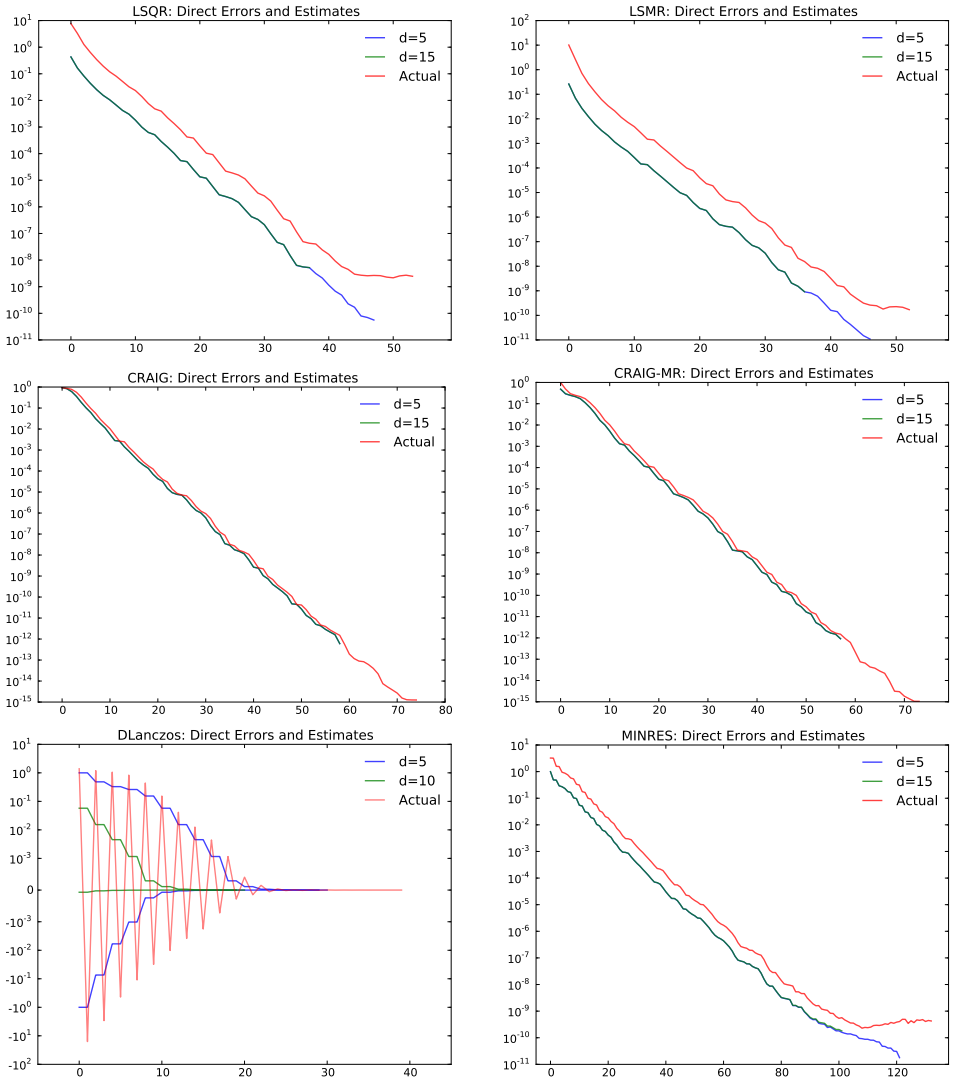

FIGURE 9.4. Lid-Driven Cavity $(578, 289)$. Note the symmetric logarithmic scale of the vertical axis for DLANCZOS used to capture the fact that the error is measured in an indefinite metric.

We insist that we have concentrated on the case where systems with the diagonal blocks $\mathbf{M}$ and $\mathbf{N}$ are easily and efficiently solved. There are clearly numerous practical cases where this assumption is not realistic. For instance, in optimization applications, $\mathbf{M}$ may represent a (possibly dense) quadratic term that is not easily inverted. It remains relatively typical that systems with $\mathbf{N}$ are easily solved. For instance in optimization $\mathbf{N}$ is usually diagonal. The same is not necessarily true in fluid flow applications where $\mathbf{N}$ may represent a mass matrix but Wathen (1987) shows that such systems are efficiently solved in a few iterations of the conjugate gradient method with diagonal preconditioner. The study of the general case,

possibly allowing for inexact solves with $\mathbf{M}$ and/or $\mathbf{N}$, is the subject of ongoing research. As a special case, this includes a finite-precision arithmetic extension of our framework.

The study of symmetric quasi-definite systems in the context of preconditioning is also the subject of ongoing research. A first important aspect is the preconditioning of SQD systems. In particular, not all preconditioners preserve the SQD structure. A second important aspect is that SQD operators may be used to precondition standard saddle-point systems, whether symmetric or not. For instance, systems encountered during the iterations of an iterative process to solve the Navier-Stokes equations typically have a zero $(2, 2)$ block. As pointed out by Benzi et al. (2005), it is possible to devise efficient SQD preconditioners for such systems.

The methods presented in this paper are relevant to optimization contexts beyond the occurence of SQD systems. Indeed, in trust-region based Gauss-Newton methods, subproblems such as (2.4) must be solved at each iteration but an accurate solution is not necessarily sought. Rather, sufficient decrease in the residual is acceptable and a constraint on the maximal norm of $\mathbf{y}$ is imposed. It is a property of standard LSQR and LSMR that the iterates $\bar{y}_k$ increase in Euclidian norm (Paige and Saunders, 1982; Fong and Saunders, 2011). Thus the quantities $\|\mathbf{y}_k\|_{\mathbf{N}}$ increase along the generalized LSQR and LSMR iterations. Consequently, it is reasonable to solve (2.4) with a trust-region constraint of the form $\|\mathbf{y}\|_{\mathbf{N}} \leq \Delta$ for some trust-region radius $\Delta > 0$ using the initial guess $\mathbf{y}_0 = 0$. As LSQR is equivalent to the conjugate gradient method on the normal equations, interrupting the iterations as soon as the boundary of the trust-region is crossed ensures sufficient decrease. In the case of LSMR, it remains necessary to establish that the decrease thus obtained is a fraction of that obtained at the Cauchy point—we refer the interested reader to (Conn et al., 2000, Chapter 7) for details.

All methods covered above apply equally to the SQD system (1.1) with $\mathbf{f} = \mathbf{0}$ and $\mathbf{g} = \mathbf{b}$. Indeed the system can be reduced to the normal equations

$$(\mathbf{A}\mathbf{N}^{-1}\mathbf{A}^{\mathsf{T}} + \mathbf{M})\mathbf{x} = \mathbf{A}\mathbf{N}^{-1}\mathbf{b},$$

which are the optimality conditions of the regularized and weighted linear least-squares problem

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \ \tfrac{1}{2} \left\| \begin{bmatrix} \mathbf{A}^{\mathsf{T}} \\ \mathbf{M}^{\frac{1}{2}} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|^2_{\mathbf{N}_+^{-1}}.$$

All methods presented in this paper are based on the lower bidiagonalization procedure referred to as "bidiag1" by Paige and Saunders (1982). A corresponding family of numerical methods may also be derived from the "bidiag2" procedure in the same paper, which performs an upper bidiagonalization of $\mathbf{A}$ and is initialized with $\mathbf{A}^{\mathsf{T}}\mathbf{b}$ instead of $\mathbf{b}$. Two variants of the Golub-Kahan process based on inner products defined by $\mathbf{M}$ and $\mathbf{N}$ and corresponding to Algorithms 4.2 and 4.3 give rise to alternative generalizations of LSQR, CRAIG, CG, LSMR, CRAIG-MR and MINRES. Whether one of those methods dominates the others numerically should be determined via intensive testing. Arioli (2010) derives a generalized variant of CRAIG based on "bidiag2" in the case where the bottom block of the matrix in (1.1) is zero, yet there exists an appropriate metric $\mathbf{N}$ to measure the norm of $\mathbf{y}$.

It does not appear possible to apply the conjugate gradient to SQD systems in general if the right-hand side does not have the form $(\mathbf{b}, \mathbf{0})$. Consider for instance

the SQD system

$$\begin{bmatrix} \mathbf{I} & \\ & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{e} \\ \mathbf{e} \end{bmatrix},$$

where $\mathbf{e}$ is the vector of all ones. At the first iteration, the conjugate gradient needs to generate the denominator

$$\begin{bmatrix} \mathbf{e}^T & \mathbf{e}^T \end{bmatrix} \begin{bmatrix} \mathbf{I} & \\ & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ \mathbf{e} \end{bmatrix} = 0$$

and must break down.

We provide stopping criteria for each method based on estimates of the relative direct error measured in the appropriate metric. It should be noted that our error estimates are not upper bounds on the actual direct error as they are measured over a window of a fixed number of iterations. Ongoing research aims to determine a cheaply-computable upper bound in the vein of Arioli (2010). For this it seems necessary to obtain a lower bound on the smallest eigenvalue in absolute value of the preconditioned operator $\bar{\mathbf{K}}$. Thanks to Theorem 5.1 an obvious lower bound is simply 1.

We gave an interpretation of the conjugate gradient method applied to a SQD system with appropriate right-hand side in terms of a min-max problem on a saddle-point function and in terms of a combination of LSQR and CRAIG. MINRES performs twice as much work as is really necessary since it aims to minimize the residual of both the normal and Schur-complement equations. Computations can be saved by employing only G-LSMR or G-CRAIG-MR.

It appears from our analysis that, on the one hand, the generalized LSQR and CRAIG, and on the other hand the generalized LSMR and CRAIG-MR are the appropriate implementations of CG and MINRES for SQD systems.

Acknowledgements. The first author wishes to express his gratitude to Emory University for allowing him to spend several sabbatical months in the mathematics and computer science department. The second author similarly wishes to express his gratitude to the Rutherford Appleton Laboratory for hosting him for several sabbatical months in the Numerical Analysis Group, where this research was born. Finally, Saunders (1995) has been a constant source of inspiration during the development of this research.

REFERENCES

M. Arioli. Generalized Golub-Kahan bidiagonalization and stopping criteria. Technical Report RAL-TR-2010-008, Rutherford Appleton Laboratory, Chilton, Didcot, Oxfordshire, UK, 2010. To appear in the SIAM Journal on Matrix Analysis.

O. Axelson and M. Neytcheva. Preconditioning methods for linear systems arising in constrained optimization problems. *Numerical Linear Algebra with Applications*, (10):3–31, 2003.

S. J. Benbow. Solving generalized least-squares problems with LSQR. *SIAM Journal on Matrix Analysis and Applications*, 21(1):166–177, 1999. DOI: 10.1137/S0895479897321830.

M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, (14):1–137, 2005. DOI: 10.1017/S0962492904000212.

H. Brézis. *Analyse fonctionnelle : théorie et applications*. Dunod, Paris, 1983.

A. Bunse-Gertner. Private communication, 2012.

R. Chandra. *Conjugate gradient methods for partial differential equations*. Ph.D. Thesis, Department of Computer Science, Yale University, New Haven CT, USA, 1978.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods*. SIAM, Philadelphia, USA, 2000.

P. Courtier. Dual formulation of four-dimensional variational assimilation. *Quarterly Journal of the Royal Meteorological Society*, 123:2449–2461, 1997. DOI: 10.1002/qj.49712354414.

J. E. Craig. The N-step iteration procedures. *Journal of Mathematics and Physics*, 34(1):64–73, 1955.

H. S. Dollar, N. I. M. Gould, W. H. A. Schilders, and A. J. Wathen. Implicit-factorization preconditioning and iterative solvers for regularized saddle-point systems. *SIAM Journal on Matrix Analysis and Applications*, 28(1):170–189, 2006. DOI: 10.1137/05063427X.

I. S. Duff. MA57—a code for the solution of sparse symmetric definite and indefinite systems. *Transactions of the ACM on Mathematical Software*, 30(2):118–144, 2004. DOI: 10.1145/992200.992202.

H. Elman, D. Silvester, and A. Wathen. *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, UK, 2005.

H. C. Elman, A. Ramage, and D. J. Silvester. Algorithm 866: IFISS, a Matlab toolbox for modelling incompressible flow. *Transactions of the ACM on Mathematical Software*, 33, 2007. DOI: 10.1145/1236463.1236469.

B. Fischer. *Polynomial Based Iteration Methods for Symmetric Linear Systems*. Number 68 in Classics in Applied Mathematics. SIAM, Philadelphia, PA, 2011. DOI: 10.1137/1.9781611971927. Originally published 1996.

D. C.-L. Fong and M. A. Saunders. LSMR: An iterative algorithm for sparse least-squares problems. *SIAM Journal on Scientific Computing*, 33(5):2950–2971, 2011. DOI: 10.1137/10079687X.

R. W. Freund, G. H. Golub, and N. M. Nachtigal. Iterative solutions of linear systems. *Acta Numerica*, 1:1–44, 1991. DOI: 10.1017/S0962492900002245.

M. P. Friedlander and D. Orban. A primal-dual regularized interior-point method for convex quadratic programs. *Mathematical Programming Computation*, 4(1): 71–107, 2012. DOI: 10.1007/s12532-012-0035-2.

A. George and K. Ikramov. On the condition of symmetric quasi-definite matrices. 21(3):970–977, 2000. ISSN 08954798. DOI: 10.1137/S0895479898333247.

A. George, K. Ikramov, and A. B. Kucherov. Some properties of symmetric quasi-definite matrices. *SIAM Journal on Matrix Analysis and Applications*, 21:1318–1323, 2000. ISSN 0895-4798. DOI: 10.1137/S0895479897329400.

P. E. Gill, M. A. Saunders, and J. R. Shinnerl. On the stability of Cholesky factorization for symmetric quasidefinite systems. *SIAM Journal on Optimization*, 17 (1):35–46, 1996. DOI: 10.1137/S0895479893252623.

I. Gohberg, P. Lancaster, and L. Rodman. *Indefinite Linear Algebra and Applications*. Birkhäuser, Basel, Switzerland, 2005.

G. H. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *SIAM Journal on Numerical Analysis*, 2(2):205–224, 1965. DOI: 10.1137/0702016.

G. H. Golub and G. Meurant. *Matrices, Moments and Quadrature with Applications*. Princeton University Press, 2010.

G.H. Golub and G. Meurant. Matrices, moments and quadrature II; how to compute the norm of the error in iterative methods. *BIT Numerical Mathematics*, 37(3): 687–705, 1997. DOI: 10.1007/BF02510247.

N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEr and SifDec, a Constrained and Unconstrained Testing Environment, revisited. *Transactions of the ACM on Mathematical Software*, 29(4):373–394, 2003. DOI: 10.1145/962437.962439.

M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.

J. Korzak. Eigenvalue relations and conditions of matrices arising in linear programming. *Computing*, 62(1):45–54, 1999. ISSN 0010-485X. DOI: 10.1007/s006070050012.

C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45(4):255–282, 1950.

C. Lanczos. Solution of systems of linear equations by minimized iterations. *Journal of Research of the National Bureau of Standards*, 49(1):33–53, 1952.

R. F. Marcia. On solving sparse symmetric linear systems whose definiteness is unknown. *Applied Numerical Mathematics*, 58(4):449–458, 2008. DOI: 10.1016/j.apnum.2007.01.014.

D. Orban. PyKrylov: Krylov subspace methods in pure Python. `github.com/dpo/pykrylov`, May 2011.

C. C. Paige. Bidiagonalization of matrices and solution of linear equations. *SIAM Journal on Numerical Analysis*, 11(1):197–209, 1974. DOI: 10.1137/0711019.

C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 12(4):617–629, 1975. DOI: 10.1137/0712047.

C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *Transactions of the ACM on Mathematical Software*, 8 (1):43–71, 1982. DOI: 10.1145/355984.355989.

I. Perugia and V. Simoncini. BlockâĂŞdiagonal and indefinite symmetric preconditioners for mixed finite element formulations. *Numer. Linear Algebra Appl.*, 7: 585–616, 2000.

Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, second edition edition, 2003. DOI: 10.1137/1.9780898718003.

M. A. Saunders. Solution of sparse rectangular systems using LSQR and CRAIG. *BIT*, 35:588–604, 1995. DOI: 10.1007/BF01739829.

D. Silvester and A. Wathen. Fast iterative solution of stabilised Stokes systems part II: Using general block preconditioners. *SIAM Journal on Numerical Analysis*, 31(5):1352–1367, 1994. DOI: 10.1137/0731070.

D. J. Silvester and V. Simoncini. An optimal iterative solver for symmetric indefinite systems stemming from mixed approximation. *Transactions of the ACM on Mathematical Software*, 37(4):42:1–42:22, 2011. DOI: 10.1145/1916461.1916466.

G. Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, Wellesley, MA, 1986.

R. J. Vanderbei. Symmetric quasi-definite matrices. *SIAM Journal on Optimization*, 5(1):100–113, 1995. DOI: 10.1137/0805005.

A. Wathen. Realistic eigenvalue bounds for the Galerkin mass matrix. *IMA Journal of Numerical Analysis*, (7):449–457, 1987. DOI: 10.1093/imanum/7.4.449.

Rutherford Appleton Laboratory, Didcot, Oxfordshire, United Kingdom.
E-mail address: mario.arioli@stfc.ac.uk

GERAD and Mathematics and Industrial Engineering Department, École Polytechnique, Montréal, Canada
URL: www.gerad.ca/~orban
E-mail address: dominique.orban@gerad.ca