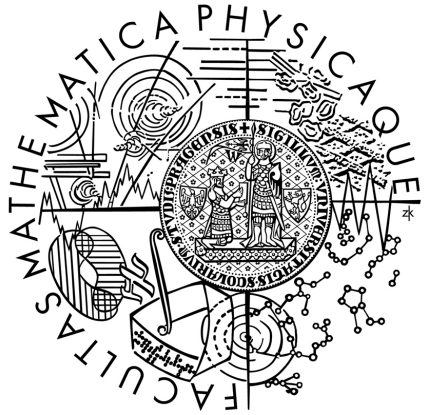
CHARLES UNIVERSITY IN PRAGUE

FACULTY OF MATHEMATICS AND PHYSICS

Habilitation Thesis



SOME RESULTS ON THE THEORY AND THE APPLICATION OF METHODS FOR SPARSE MATRICES

Jurjen Duintjer Tebbens

Mathematics - Section: Mathematical modeling and numerical mathematics

Prague, September 2015

webpage: http://www.cs.cas.cz/duintjertebbens
e-mail: duintjertebbens@cs.cas.cz

Contents

Preface **5**

1 Introduction **7**
1.1 On the convergence of Krylov subspace methods 12
1.2 Condition number estimation 16
1.3 Preconditioning linear system sequences 18
1.4 Classification with high-dimensional data 21

Bibliography **25**

2 Convergence analysis of Krylov subspace methods for non-normal matrices **36**
J. Duintjer Tebbens, G. Meurant: *Any Ritz value behavior is possible for Arnoldi and for GMRES*, SIAM Journal on Matrix Analysis and Applications, vol. 33, no. 3, pp. 958–978, 2012. . . 36
J. Duintjer Tebbens, G. Meurant, H. Sadok, Z. Strakoš: *On investigating GMRES convergence using unitary matrices*, Linear Algebra and its Applications, vol. 450, pp. 83–107, 2014. 57
G. Meurant, J. Duintjer Tebbens: *The role eigenvalues play in forming GMRES residual norms with non-normal matrices*, Numerical Algorithms, vol. 68, pp. 143-165, 2015. 82

3 Incremental condition number estimation **105**
J. Duintjer Tebbens, M. Tůma: *On incremental condition estimators in the 2-norm*, SIAM Journal on Matrix Analysis and Applications, vol. 35, no. 1, pp. 174–197, 2014. 105

4 Updated preconditioners for sequences of linear systems **129**
J. Duintjer Tebbens, M. Tůma: *Efficient preconditioning of sequences of nonsymmetric linear systems*, SIAM Journal on Scientific Computing, vol. 29, no. 5, pp. 1918–1941, 2007. 129
Ph. Birken, J. Duintjer Tebbens, A. Meister, M. Tůma: *Preconditioner updates applied to CFD model problems*, Applied Numerical Mathematics, vol. 58, no. 11, pp. 1628–1641, 2008. . 153

J. Duintjer Tebbens, M. Tůma: *Preconditioner updates for solving sequences of linear systems in matrix-free environment*, Numerical Linear Algebra with Applications, vol. 17, pp. 997–1019, 2010. 167

5 Classification of high-dimensional data with Fisher's linear discriminant analysis 190

J. Duintjer Tebbens, P. Schlesinger: *Improving implementation of linear discriminant analysis for the high dimension/small sample size problem*, Computational Statistics and Data Analysis, vol. 52, no.1, pp. 423–437, 2007. 190

Preface

This habilitation thesis presents some selected results related to sparse matrix methods. They range from theoretical results addressing convergence analysis of Krylov subspace methods to practical considerations on the application of sparse eigensolvers for the statistical task of classification, while an important part is about efficient preconditioning. The first chapter briefly introduces the reader to the broad research area of sparse matrix methods and in subsequent subsections to the more specialized results presented in the remaining chapters. Chapters 2 till 5 display the full text of the following 8 publications.

Chapter 2: Convergence analysis of Krylov subspace methods for non-normal matrices.

1. J. Duintjer Tebbens, G. Meurant: *Any Ritz value behavior is possible for Arnoldi and for GMRES*, SIAM Journal on Matrix Analysis and Applications, vol. 33, no. 3, pp. 958–978, 2012.

2. J. Duintjer Tebbens, G. Meurant, H. Sadok, Z. Strakoš: *On investigating GMRES convergence using unitary matrices*, Linear Algebra and its Applications, vol. 450, pp. 83–107, 2014.

3. G. Meurant, J. Duintjer Tebbens: *The role eigenvalues play in forming GMRES residual norms with non-normal matrices*, Numerical Algorithms, vol. 68, pp. 143-165, 2015.

Chapter 3: Incremental condition number estimation.

4. J. Duintjer Tebbens, M. Tůma: *On incremental condition estimators in the 2-norm*, SIAM Journal on Matrix Analysis and Applications, vol. 35, no. 1, pp. 174–197, 2014.

Chapter 4: Updated preconditioners for sequences of linear systems.

5. J. Duintjer Tebbens, M. Tůma: *Efficient preconditioning of sequences of nonsymmetric linear systems*, SIAM Journal on Scientific Computing, vol. 29, no. 5, pp. 1918–1941, 2007.

6. Ph. Birken, J. Duintjer Tebbens, A. Meister, M. Tůma: *Preconditioner updates applied to CFD model problems*, Applied Numerical Mathematics, vol. 58, no. 11, pp. 1628–1641, 2008.

7. J. Duintjer Tebbens, M. Tůma: *Preconditioner updates for solving sequences of linear systems in matrix-free environment*, Numerical Linear Algebra with Applications, vol. 17, pp. 997–1019, 2010.

Chapter 5: Classification of high-dimensional data with Fisher's linear discriminant analysis.

8. J. Duintjer Tebbens, P. Schlesinger: *Improving implementation of linear discriminant analysis for the high dimension/small sample size problem*, Computational Statistics and Data Analysis, vol. 52, no.1, pp. 423–437, 2007.

In the first chapter, citations of publications where the author of this habilitation thesis is co-author are marked with a lower index DT, e.g. $[1]_{DT}$ means that the author of the present habilitation thesis is coauthor of publication [1].

For their support I would like to thank in the first place Zdeněk Strakoš and Miroslav Tůma, then my collaborators, in particular Gérard Meurant, Philipp Birken, Andreas Meister, Pavel Schlesinger and Jan Kalina, my colleagues at the Department of Computational Methods and, above all, my family.

Jurjen Duintjer Tebbens
Prague, September 2015.

Chapter 1

Introduction

Let us consider a *sparse* real matrix A of size n,

$$A \in \mathbb{R}^{n \times n}.$$

We here assume that the matrix A is real, but all results of the habilitation thesis can be easily generalized for complex matrices as well. Sparsity of a matrix is often defined as the property that the number of zero entries is high enough to take advantage of in arithmetical manipulations and software implementations [120, Chapter 3], without further quantitative specification of the number of zero entries. A large number of methods can take advantage of or are specially designed to take advantage of sparsity of matrices and vectors (see, e.g., [138, Chapter2] [139, Chapter 3]) ; we will discuss here only the methods relevant for this thesis. A standard operation where sparsity can be exploited is matrix-vector multiplication, with the number of required floating point operations (flops) being proportional to the number of nonzeros of A. They can be of the order n when only a few diagonals of A are nonzero, whereas multiplication with a *dense* (i.e. non-sparse) matrix asks for $\mathcal{O}(n^2)$ flops in general. The popular class of Krylov subspace methods consists of methods based on repeated multiplication of vectors with a given matrix (see, e.g., [82] or [113]) and is therefore especially suited for large sparse matrices. Direct solvers which compute matrix decompositions [80] like an LU decomposition are somehow less naturally favorable for sparsity, because the decompositions can be dense even with a sparse input matrix. The aim therefore is to keep the decomposition maximally sparse (i.e. to avoid *fill-in*) by using for instance suitable row or column permutations(see, e.e., [40, 120]). Sparse direct solvers often rely on results from graph theory and apply them to a graph representing the sparsity pattern of A (see, e.g., [120, 69]). Graph theory plays an important role as well in partitioning sparse matrices (or also sparse vectors) for efficient storage and parallel implementations. For *in*complete matrix decompositions, which are used to accelerate Krylov subspace methods, it is customary to neglect (*drop*) small size entries (see, e.g., [139, Chapter 10]) and thus enforce sparsity of the computed factors.

In this habilitation thesis we focus on sparse matrix methods for solving linear systems of algebraic equations, while some results concern eigenvalue problems. Consider the solution of a linear system

$$Ax = b, \qquad b \in \mathbb{R}^n, \tag{1.1}$$

with a non-singular, sparse and possibly very large matrix A. Sparse linear solvers may be roughly divided into three classes, which are often combined in practical computations: Krylov subspace methods, direct (LU-based) solvers and multigrid-type methods (including domain decomposition, Schwarz and hierarchically semiseparable matrix methods). Multigrid methods use multiple, sometimes fictitious, discretization levels and combine several techniques on the different levels (see, e.g. [142]). For some important classes of problems they are asymptotically optimal, but they can also be sensitive to changes of the problem [62]. Modern LU packages (e.g. UMFPACK [41], GPLU [75], PARDISO [141], MUMPS [2] or WSMP [88]) use heuristics based on graph theory and combinatorics to preserve sparsity and use advanced developments in computer science like BLAS, paralell or out-of-core implementation techniques [76]. They are able to solve huge sparse linear systems (with billions of unknowns) in very short CPU-time. However, if fill-in cannot be controlled efficiently, the L and U factors become sometimes too large to store for the available computer memory. An advantage of Krylov subspace methods is that they are efficient for a wide variety of problems. They are iterative and thus allow the computation to be stopped as soon as a satisfactory approximation to the solution has been found. Depending on the application, the required approximation needs not be very accurate. In addition, they allow *matrix-free* implementations, where the multiplication of A with vectors is done without storing A. For example, it can be performed using a difference scheme if A represents the Jacobian matrix of a function to be minimized. In most Krylov subspace methods multiplication with A is the only information needed about A (but some methods require multiplication with the transposed of A as well). Then the system matrix needs not be stored at all, leading to considerable memory savings.

The most important part of this habilitation thesis adresses Krylov subspace methods and their acceleration through preconditioning. A Krylov subspace for A and a given vector v (not necessarily the right-hand side b in (1.1)) is a subspace generated by the vectors $v, Av, A^2v, A^3v, \ldots$ (for an overview see, e.g., [105, 154, 82, 113]). More precisely, the kth Krylov subspace $\mathcal{K}_k(A, v)$ for A and v is defined as

$$\mathcal{K}_k(A, v) \equiv \text{span}\{v, Av, ..., A^{k-1}v\}.$$

Krylov subspace methods are based on projection of the given, large matrix problem for A (e.g., a linear system, eigenproblem, matrix function problem) to Krylov subspaces of small dimension which grow with the iteration number. In favorable cases, dominant properties become apparent after a relatively small

number of iterations and may yield sufficiently accurate approximations to the solution. Convergence analysis, the first topic of this thesis, studies how fast approximations with a given accuracy are found depending on properties of A and v. We remark that even if we consider linear systems and linear eigenvalue problems, convergence analysis consists of highly nonlinear problems due to the structure of the Krylov subspaces [149] (all vectors in $\mathcal{K}_k(A, v)$ are of the form $\pi(A)v$ for a polynomial π of degree $k-1$). Focusing for a moment on linear systems, this is in contrast with the situation for classical, so-called stationary iterative methods based on a matrix splitting like Gauß-Seidel or SOR [156]. Convergence analysis for these methods traditionally studies the asymptotic convergence factor, i.e. the factor by which the Euclidean norm of the *residual vector*,

$$r_k = b - Ax_k,$$

where x_k is the approximation obtained in the kth iterate, is reduced in every iteration [156]. In other words, it investigates the linear convergence rate. In applications where Krylov subspace methods are most often used, however, one hopes to find acceptable approximations after a rather small number of iterations already. Hence what is important is not the asymptotic linear convergence, but fast, usually superlinear convergence during the initial phase of the iterative procedure [152, 155]. As we explain below, convergence analysis is particularly challenging for non-normal matrices A, which probably account for the majority of cases arising in real-world applications. A matrix A is *normal* if it commutes with its transposed matrix denoted by A^T, i.e. if $AA^T = A^TA$.

Although in some applications, Krylov subspace methods indeed yield satisfactory approximations in a low number of iterations, in general some type of acceleration needs to be added for fast convergence. The most popular acceleration technique for solving linear systems is probably preconditioning. An efficient left *preconditioner* M for (1.1) is a matrix relatively cheaply computable from A, allowing cheap solution of linear systems with that matrix and such that the solution process for

$$M^{-1}Ax = M^{-1}b$$

with a Krylov subspace method converges considerably faster than if the Krylov subspace method is applied to the original system (1.1) (for an overview see, e.g., [37] or [13]). A right preconditioner satisfies the same criteria except for that it modifies the original problem to the linear system

$$AM^{-1}y = b,$$

so that the solution of the original system (1.1) is $x = M^{-1}y$. Preconditoners modifying (1.1) from both the left and the right are used as well, in particular for symmetric linear systems in order to preserve symmetry. The term preconditioning originates from the solution of symmetric linear systems, where a reduction of the 2-norm *condition number* of the system matrix can be expected

to improve the convergence speed of the used Krylov subspace method [139, Chapter 6]. The 2-norm condition number $\kappa(A)$ of a nonsingular matrix A is defined as

$$\kappa(A) = \|A\| \cdot \|A^{-1}\|,$$

where $\|\cdot\|$ denotes the Euclidean norm. The estimation of condition numbers is treated in detail in Section 1.2. For symmetric matrices, the 2-norm condition number equals the ratio

$$\kappa(A) = \frac{|\lambda_1|}{|\lambda_n|},$$

where $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n| > 0$ are the (real) eigenvalues of A. The way many preconditioners attempt to improve convergence for symmetric linear systems is through eigenvalue clustering to reduce the 2-norm condition number. We note that modification of the eigenvalues of A by the choice of an appropriate preconditioner can be useful for non-normal matrices as well, but in general the spectrum of non-normal system matrices needs not decide about the converge speed of the used Krylov subspace method [84]. This is the main topic of Section 1.1 below.

A popular class of preconditioners is given by the incomplete LU factorizations mentioned earlier. They approximate A with a product $M = LU$, where L is a lower and U an upper triangular matrix. The solution of linear systems with M requires a forward solve with L and a backward solve with U, both requiring $\mathcal{O}(n^2))$ flops at most (for sparse factors L and U the costs are in general much lower). The computation of the incomplete factorization itself has computational costs proportional to the number of nonzeros of A if a standard dropping strategy is used [139, Chapter 10]. With a reasonable accuracy of the incomplete factorization, the system matrix of the preconditioned linear system may approximate the identity matrix well enough to yield faster convergence than for the original system when a Krylov subspace method is applied.

A different class of preconditioners attempts to find explicit cheap approximations of A^{-1}, instead of A. In other words, one constructs M^{-1} such that $M^{-1} \approx A^{-1}$ instead of constructing M such that $M \approx A$. This has the advantage that multiplication of vectors with the preconditioned system matrix needed in Krylov subspace methods does not require the solution of a linear system, but only multiplication with the explicitly constructed matrix M^{-1}. This can be easily done in parallel, whereas the solution of triangular systems needed with incomplete LU factorizations is much more difficult to parallelize [13]. On the other hand, approximations of A^{-1} tend to be less sparse than approximations of A if A itself is sparse. Preconditioners of this class (approximate inverse preconditioners), include the SPAI [86] and AINV [16, 17] preconditioners.

Incomplete and approximate inverse factorizations belong to the most robust and universally applicable preconditioners. Physics-based preconditioners

take into account the specific physical problem underlying the given linear system [104]. For instance, if the linear system arises from the discretization of a convection-diffusion or convection-diffusion-reaction equation, then the preconditioner may consist of the matrix for the discretization corresponding to the diffusion part only. Physics-based preconditioners are mostly available in matrix-free form, which is not the case for the previously mentioned types of preconditioners. Another matrix-free preconditioning technique consists of applying a number of inner iterations with a suitable Krylov subspace method (which need not be the same as the method used for outer iterations).

A popular acceleration technique for linear systems, which can but need not be incorporated in preconditioning, is deflation (see [125, 103, 36, 64, 35, 116, 77, 78], to cite just some of the proposed approaches and a few overviews). The main idea is to use information about approximate eigenspaces of A, computed along with the iterative process of the Krylov subspace method, to eliminate the influence of eigenvalues assumed to hamper convergence.

Section 1.3 of this chapter addresses the preconditioning of *sequences* of linear systems. Linear system sequences arise in a wide variety of applied mathematics. Some examples are simulation of processes in fluid dynamics where every time step requires the solution of a system of nonlinear equations, operation research problems where the linear programs are solved with the simplex method, the solution of Helmholtz equations for the propagation of time-harmonic waves, structural mechanics problems or numerical optimization.

Preconditioning is a research area probably more closely related to practice than the theory of iterative methods like Krylov subspace methods. One sometimes hears the phrase that preconditioning is "a combination of art and science" [139, Chapter 10]. It is often hard to predict a priori the efficiency of a preconditioner which involves its construction costs, its application costs and the resulting number of iterations of the Krylov subspace method. Particular applications usually ask for tailor-made preconditioners whereas Krylov subspace methods can be applied to basically all type of problems involving large sparse matrices.

The field where both preconditioning and Krylov subspace methods were first applied, is finite element and difference discretization of partial differential equations. But in recent years they have shown to be useful in many other research area's where sparse matrices arise. The last topic of the habilitation thesis concerns the somehow remote field of statitistics (a field that also can be seen as a combination of art and science). Section 1.4 introduces the statistical task of data classification, an application in which efficient operations with sparse matrices start to become very important.

Throughout the thesis computations are assumed to run in exact arithmetic. In practice finite precision arithmetic is used and although it can significantly influence computations, we do not consider issues related to finite precision computations here.

1.1 On the convergence of Krylov subspace methods

Methods for symmetric matrices occupy a special place among Krylov subspace methods. The most popular Krylov subspace methods for symmetric matrices are the Conjugate Gradient (CG) method [92, 109] for positive definite linear systems, the MINRES method [130] for indefinite systems and the Lanczos method [108] for symmetric eigenproblems. They build bases for the involved Krylov subspaces that are orthogonal, a property which is very desirable with respect to numerical stability and implementation. With symmetric matrices, they can be constructed using three-term recurrences by an orthogonalization method called Lanczos process [108, 109]. This implies low, constant computational costs per iteration as well as low memory requirements.

Symmetric matrices also lead to an analysis of convergence behavior which is well understood. For instance when using the MINRES method to solve an indefinite symmetric linear system, the iterate x_k in the kth iteration minimizes, with initial guess $x_0 = 0$, the norm of the kth residual vector $r_k = b - Ax_k$ over all vectors in the kth Krylov subspace $\mathcal{K}_k(A, b)$. Therefore, residual norms are non-increasing and satisfy

$$\|r_k\| = \min_{p \in \pi_k} \|p(A)b\|, \tag{1.2}$$

where π_k is the set of polynomials of degree k with the value one at the origin. Because A is symmetric, it can be decomposed as

$$A = V \Lambda V^T, \quad \Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n), \quad V^T V = I_n, \tag{1.3}$$

where I_n denotes the size n identity matrix. The entries $\lambda_1, \ldots, \lambda_n$ of the matrix Λ are the eigenvalues of A and the columns of V are the corresponding eigenvectors. Substituting this decomposition into (1.2) gives

$$\|r_k\| = \min_{p \in \pi_k} \|p(\Lambda) V^T b\|, \tag{1.4}$$

showing that MINRES residual norms are fully determined by two quantities: eigenvalues and components of the right-hand side in the eigenvector basis. A closed-form expression for the kth MINRES residual norm in terms of these quantities, i.e. the solution of (1.4), was presented in [9] and in $[52]_{DT}$ (it generalizes the results in [115] for the special case $k = n - 1$). From (1.4), the upper bound

$$\frac{\|r_k\|}{\|b\|} \leq \min_{p \in \pi_k} \max_{i=1,\ldots,n} |p_k(\lambda_i)|, \tag{1.5}$$

can be derived (see, e.g., [139, Chapter 6]). It consists of a min-max approximation problem which depends on the spectrum *only*. The bound is sharp in

the sense that for every k there exists a right-hand side (depending on k) such that equality holds. For the CG method, (1.4) and (1.5) hold with the 2-norm replaced by the A-norm and the residual replaced with the error $e_k = x - x_k$ ($\|b\| = \|r_0\|$ is replaced with e_0). For A a symmetric positive definite matrix, the A-norm $\|v\|_A$ of a vector v is defined as $\sqrt{v^T A v}$. A well-known bound [139, Chapter 6] for CG derived from the CG variant of (1.5) and involving the condition number of A is

$$\frac{\|e_k\|_A}{\|e_0\|_A} \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k .$$

This bound is often used to explain the improved convergence of CG when a preconditioner that clusters eigenvalues is applied.

Krylov subspace methods for non-symmetric matrices are more complicated in several ways. First, they generally do not allow the construction of orthogonal bases with short recurrences. More precisely, unless the matrix belongs to the set of normal(s) matrices, which is a subset of normal matrices, orthogonal bases for $\mathcal{K}_k(A, v)$ cannot be generated (for any initial vector v) with $(s+2)$-term recurrences [67]. For the definition of normal(s) and discussions of this fundamental result we also refer to [112, 66] and [113, Chapter 3]. Thus with non-normal matrices, if we wish to use orthogonal bases of Krylov subspaces, we have to generate them with long recurrences whose computational and storage costs grow with the iteration number. This is done in the GMRES [140] and FOM [136, 137] methods to solve linear systems and in the Arnoldi method for eigenproblems [135]. All three methods use the Arnoldi orthogonalization process to build orthogonal bases. If on the other hand, we wish to use short recurrences with non-normal matrices, we must give up orthogonality. Often it is replaced with bi-orthogonality of a pair of bases, one for $\mathcal{K}_k(A, v)$ and one for $\mathcal{K}_k(A^T, w)$ with a shadow vector w. Popular methods for non-normal linear systems using short recurrences include Bi-CG [71], QMR [72] and Bi-CGSTAB [153]. A different, straightforward way to bound the length of recurrences is through restarting a long-recurrence method after a small number of iterations; the most popular method of this type is restarted GMRES.

As for convergence analysis of Krylov subspace methods for non-normal linear systems, despite considerable efforts, it is in fact not well understood [114]. The properties of A and v that dominate convergence behavior are not clear. These are *not* the distribution of eigenvalues and the components of the right-hand side in the eigenvector basis, as is the case for symmetric matrices: When A is not normal, there is no orthogonal eigenvector basis and the decomposition (1.3) must replaced with

$$A = V J V^{-1}, \tag{1.6}$$

where in the worst case, for non-diagonalizable matrices, J is a bidiagonal Jordan matrix. If we concentrate on the GMRES method, which satisfies the

same optimality criterion (1.2) as the MINRES method, we see that substituting (1.6) in (1.2) does not reduce to the minimization problem (1.4) but to

$$\|r_k\| = \min_{p \in \pi_k} \|V\, p\,(J) V^{-1} b\|, \tag{1.7}$$

which is much more difficult to analyze and which shows that the eigenvectors (or principal vectors) play a more important role. The interplay between eigenvalues, eigenvectors and right-hand side seems to be rather complicated. This was confirmed in $[122]_{DT}$ (the last paper of Chapter 2 of this habilitation thesis) which presents the solution of (1.7), thus giving a complete description of how eigenvalues contribute in forming residual norms and of what quantities can prevent GMRES from being governed by eigenvalues.

Many papers look for approaches other than eigenvalue analysis to explain GMRES convergence, possibly in an elegant way. They include approaches based on pseudospectra [151, 126], the field of values [58], the polynomial numerical hull [83], potential theory [107], decomposition in normal plus low-rank [98] or comparison with GMRES for non-Euclidean inner products [133]. Though they can be very suited to explain convergence for particular problems, none of the approaches seems to represent a universal tool for GMRES analysis. In [155] it was suggested that convergence of the eigenvalue approximations generated in the Arnoldi orthogonalization process, the *Ritz values*, to eigenvalues of A will often explain the acceleration of convergence of GMRES.

The probably most convincing results showing that GMRES needs not be governed only by eigenvalues can be found in a series of papers by Arioli, Greenbaum, Pták and Strakoš [85, 84, 4]. They represent an alternative approach to gain insight in convergence behavior, namely through investigation of the form taken by matrices (with initial vectors) that generate the same convergence behavior. The first paper [85] showed that if a residual norm convergence curve is generated by GMRES, the same curve can be obtained with a matrix having prescribed nonzero eigenvalues (an analogue on prescribed nonzero singular values can be found in [59]). In addition, the same curve can be generated with a normal and even with a unitary matrix. This naturally leads to the question whether some properties of A and b can be related to properties of a unitary matrix V generating the same residual norms as A and b. Spectral properties of V are particularly interesting because they *do* govern convergence behavior since V is normal. An answer pointing out, once more, the indispensable role of eigenvectors, was given in $[52]_{DT}$, which forms the second part of Chapter 2.

Greenbaum, Pták and Strakoš [84] complemented the earlier results of [85] by proving that *any* nonincreasing sequence of residual norms can be generated by the GMRES method (a similar result for residual norms at the end of restart cycles in the restarted GMRES method can be found in [157]). Furthermore, in Arioli, Pták and Strakoš [4] a complete parametrization was given of all pairs $\{A, b\}$ generating a prescribed residual norm convergence curve and such that

A has prescribed spectrum. The conclusion of this paper mentions that it is desirable to formulate similar parametrizations for the early termination case (i.e. the situation where GMRES finds, in exact arithmetic, the solution of the linear system in less than n iterations). Some aspects of the early termination case related to the minimal polynomial were pointed out in the next to last section of that paper. Related results for early termination were described in the Ph.D. thesis of Liesen [110]; see also [111]. In $[51]_{DT}$, we gave a complete parametrization of all matrices and right-hand sides yielding a prescribed non-increasing GMRES convergence curve terminating before iteration n and where the input matrix has prescribed nonzero eigenvalues.

The residual norms generated in the GMRES method can be described, theoretically, by the particularly simple and natural minimization property (1.2). This is perhaps the main cause of the fact that the majority of convergence results about Krylov methods for nonsymmetric matrices concern the GMRES method (and to a lesser extent FOM). The analysis of methods like Bi-CG, QMR and Bi-CGStab is further complicated by several types of potential breakdowns, non-orthogonal projection processes and the presence of a second Krylov subspace, $\mathcal{K}_k(A^T, w)$. In $[49]_{DT}$ an attempt was made to generalize convergence results like [85, 84, 4] to some methods which do not employ orthonormal bases like Bi-CG and QMR. The paper shows that, just as for GMRES, linear systems can be constructed with arbitrary spectrum and generating arbitrary residual norms when these methods are applied.

A similarly strong division between convergence theory for normal and non-normal matrices exists for eigenproblems solved with Krylov subspace methods. A fundamental tool in the convergence analysis of the Lanczos method for symmetric eigenproblems is the *interlacing property* for the eigenvalues of the subsequently generated tridiagonal matrices (the restrictions of A to small Krylov subspaces). The interlacing property is instrumental for the persistence theorem on stabilization of Ritz values (see, e.g., [127, 128, 129], [123] or [48, Chapter 7] (in Czech)). Several generalizations of the interlacing property to normal matrices exist; see e.g. [68, 3], or the publications [118, 65] with geometric interpretations. However, potentially non-normal input matrices make convergence analysis of the Arnoldi method for non-normal eigenproblems delicate. There is no interlacing property for the principal submatrices of general non-normal matrices, see [144] for a thorough discussion on this topic and its relation to the field of Lie algebra's.

The GMRES and the Arnoldi methods being closely related through the Arnoldi orthogonalization process, a naturally arising question is whether a result, similar to the results of Arioli, Greenbaum, Pták and Strakoš, on arbitrary convergence behavior of the Arnoldi method can be proved. By arbitrary convergence behavior of the Arnoldi method we mean the ability to prescribe *all* Ritz values from the very first until the very last iteration. The opening paper of Chapter 2 (i.e., the publication $[50]_{DT}$) gives an affirmative answer and a parametrization of the class of all matrices and initial Arnoldi vectors that gen-

erates prescribed Ritz values in all iterations. Besides this result on arbitrary convergence behavior of the Arnoldi method, it derives a parametrization that allows to characterize all pairs $\{A, b\}$ generating arbitrary convergence behavior of *both* GMRES and Arnoldi. In this sense not even Ritz values generated in the GMRES method do, in general, have an influence on the generated residual norms.

The negative results of $[50]_{DT}$ can save researchers lots of effort. It is not possible to prove anything on the convergence speed of Arnoldi's method without special assumptions on A (and on the initial vector v). In practice one uses specialized variants like implicit restarts and exact shifts [6, 7], making convergence analysis even more complicated with some examples of non-convergence given in [63]. A discussion on the impact of our results for practice can be found in the conclusion of $[50]_{DT}$. Les us mention one consequence for the solution of non-symmetric linear systems. Deflation techniques are popular to accelerate the convergence of (restarted) GMRES. As explained earlier, they try to eliminate eigenvalues that are supposed to slow down the convergence. There are two potential problems with such a strategy: First, any convergence speed is possible with any nonzero eigenvalues. Second, information on the eigenvalues of A must be obtained from the Arnoldi orthogonalization process, i.e. basically from the Arnoldi method. But Ritz values need not converge to the eigenvalues at all. Though one frequently uses harmonic Ritz values and in practice, deflation is often beneficial, there is no sound theoretical explanation for the efficiency of deflation methods for non-normal linear systems.

We close this section, which discussed several results on constructing systems with prescribed convergence behavior, with the remark that in $[46]_{DT}$ and $[45]_{DT}$ the author proposed a way to improve restarted GMRES through switching to a rank-one updated linear system with prescribed residual norms when GMRES is applied.

1.2 Condition number estimation

We have seen that the condition number $\kappa(A) = \|A\|\|A^{-1}\|$ is a quantity that can be useful for convergence bounds of Krylov subspace methods. It is in fact a very important tool to assess in various ways the quality of computed solutions of linear algebraic equations and eigenvalue problems and their sensitivity to perturbations. For example, if the right-hand side in (1.1) is perturbed by a vector Δb, then the resulting perturbation Δx of the solution in

$$A(x + \Delta x) = b + \Delta b$$

can be related to Δb through

$$\frac{\|\Delta x\|}{\|x\|} \leq \kappa(A) \, \frac{\|\Delta b\|}{\|b\|}.$$

The condition number is in general rather expensive to compute because it requires knowledge about the inverse of the matrix (as is clear from its definition). In fact, its computation costs are often of the same order as for the solution of the corresponding linear system or eigenproblem. Computation of the 2-norm condition number is particularly expensive because the matrix 2-norm is itself expensive. Efficient but cheap condition number *estimates* are highly desirable. Not surprisingly, the most popular condition number estimators approximate the easily computable 1-norm (or column norm) condition number, see, e.g., [89], [94], [95]. The method proposed in [96], based on [89], [94], [95], is the standard condition estimator implemented in Matlab [119].

Estimators for the Euclidean norm (2-norm) condition number are, however, possibly needed more often than for any other norm. Traditionally they are computed with the help of a triangular decomposition of A [93] (as are, when necessary, matrix inverses), like an LU decomposition or if the matrix is symmetric positive definite, a Cholesky decomposition. The estimator then estimates the condition numbers of the triangular factors, which is more efficient than estimation for an unstructured original matrix. A breakthrough in the development of estimators in the 2-norm was *incremental condition estimation* of a triangular matrix. This way of estimation is closely related to the triangular decomposition process and to incomplete triangular decompositions, as will be explained below.

Incremental condition estimation was proposed by Bischof at the beginning of the nineties [24], [25] and further generalized for solving related tasks [26], [145]. It computes approximate condition numbers for consecutively all leading upper left submatrices of the given triangular matrix, starting from the smallest submatrix of size one. The estimate for the current submatrix is obtained from an approximate left singular vector constructed through updating the approximate left singular vector for the previous submatrix, without however accessing it. This makes the procedure relatively inexpensive; the costs for incremental condition estimation are of order n^2 for dense matrices. It is particularly suited when a triangular matrix is computed one row or column at a time, which is precisely what is usually done during the LU decomposition process. An analogue strategy based on approximate *right* singular vectors was proposed later by Duff and Vömel [44] and recommended for sparse matrices, while it appears to do worse than [24] for the estimation of the factor $\|A^{-1}\|$ in the definition of the condition number. Both incremental estimators compute lower bounds which are in general within a factor 2 to 10 from the exact condition number.

Chapter 3 (i.e., the publication $[57]_{DT}$) shows that an appropriate combination of the technique based on right singular vectors [44] with cheap inversion of the involved triangular matrix leads to an incremental condition estimator which is significantly more accurate than the estimators of [24, 44]. It obtains lower bounds which are on average within a factor 1.2 from the exact condition number. The costs of triangular matrix inversion are low compared to those

of the triangular decomposition procedure itself. Moreover, some variants of decomposition compute not only the triangular factors, but simultaneously, as a by-product, the inverses of these factors [31, 32, 33]. In *in*complete factorization, information on the inverses of the progressively computed factors is important to guarantee robust dropping and pivoting (permutation) rules. For instance, the robust incomplete LU decompositions implemented in the popular software package ILUPACK [29] estimate the infinity norm (row-norm) condition number of submatrices to control the growth of the entries of inverse submatrices [27, 28]. The usage of our improved 2-norm estimators could lead to a further increase of robustness. Future work with respect to the new estimator from $[57]_{DT}$ include a variant where the inverse factors need not be stored (which is particularly important for sparse matrices whose inverses tend to be dense) and incremental condition estimation of preconditioned system matrices.

1.3 Preconditioning linear system sequences

A natural property of most sequences of linear systems arising in the applications mentioned earlier, is that the individual linear systems change relatively slowly in the course of the sequence. One therefore often attempts to reuse information gained from the solution of one linear system for the solution of some of the following systems. Some examples are the usage of hot starts (the solution of the previous system or an extrapolation obtained from the solutions of several previous systems), recycling of Krylov subspaces generated earlier [131, 116] or recycling of spectral information [77]. A different way to save costs throughout the sequence is through using efficient stopping criteria [60]. In some cases exact updating of the system matrix is feasible even for large problems. Rank-one updates of LU factorizations have been used since decades in the simplex method where the change of one system matrix to another is restricted to one column [150]. General rank-one updates of an LU decomposition are discussed in [148].

For many difficult problems, the linear systems of the sequence need to be preconditioned with a powerful type of incomplete factorization, which is typically relatively expensive to compute. A popular straightforward strategy to share part of the computational effort throughout the sequence is freezing of the preconditioner for a number of linear systems of the sequence. Let us consider for a moment one of the standard tasks where linear system sequences arise, function optimization. A system of nonlinear equations $F(x) = 0$ for a non-linear function $F : \mathbb{R}^n \to \mathbb{R}^n$ solved by a Newton- or Broyden-type method leads to a sequence of problems of the form

$$A^{(i)}(x_{i+1} - x_i) = b^{(i)},\ i = 1, \ldots, \tag{1.8}$$

where the right-hand side $b^{(i)}$ equals $-F(x_i)$ and the matrix $A^{(i)}$ is the Jacobian evaluated in the current iteration x_i of the Newton process or its approximation

[101], [102]. A well-known strategy when solving with a Newton-type method is to skip evaluations of the (approximate) Jacobian during some iterations, leading to Shamanskii's combination of the chord and Newton method [30], [143]. Hence during several subsequent systems of the sequence the system matrix is frozen. The systems differ only by their right hand side and linear solving techniques with multiple right hand sides can be exploited [146], [74]. An alternative way to save costs is through freezing the preconditioner only, i.e. allowing the changing of the system matrices [104].

In many situations a frozen preconditioner can be very efficient for a large portion of the sequence (see, e.g., $[90]_{DT}$.) In other applications, as the sequence progresses the number of iterations to solve linear systems with a frozen preconditioner tends to deteriorate. To enhance the power of frozen preconditioners one may in addition use approximate preconditioner *updates*. In Quasi-Newton methods the difference between system matrices is of small rank and preconditioners may be efficiently adapted with approximate small-rank updates; this has been done in the symmetric positive definite case, see [19, 124]. In a more general situation, however, it is not clear whether the system matrices change in a structured way and, in fact, it is not clear either what an *exact* update of a preconditioner means. Let us try to explain the idea of approximate preconditioner updates more in detail.

In order to simplify the notation, consider two linear systems of the sequence, one reference system denoted by $Ax = b$ and a system arising later in the sequence denoted by $A^+x^+ = b^+$. In many situations, the individual systems of a sequence are not available simultaneously, but the systems follow from the solution of previous systems. Let us denote the difference matrix $A - A^+$ by B and let M be a reference preconditioner approximating A. Some information about the quality of the preconditioner M can be taken from the distance

$$\|A - M\|_N \tag{1.9}$$

for an appropriate matrix norm $\|\cdot\|_N$ or from the distances

$$\|I - M^{-1}A\|_N \quad \text{or} \quad \|I - AM^{-1}\|_N \tag{1.10}$$

depending on whether we precondition from the left or right (see, e.g. [14]). If preconditioners are in factorized form, both (1.9) and (1.10) should be considered in practice since the preconditioners can suffer from two types of deteriorations. While the norm of the matrix (1.9) expresses *accuracy* of the preconditioner, the norms of the matrices (1.10) relate to its *stability* [39], see also [15]. We immediately obtain

$$\|A - M\|_N = \|A^+ - (M - B)\|_N.$$

Hence

$$M^+ \equiv M - B$$

represents an updated preconditioner for A^+ of the same level of accuracy as M represents for A. This updated preconditioner may be regarded as an exact update with respect to accuracy.

If we want to use M^+ as a preconditioner, we need to multiply vectors with its inverse in every iteration of the linear solver. In some problems, the difference matrix B is such that $(M-B)^{-1}$ can be obtained from M^{-1} with low costs. For instance if B has small rank, M^+ can be easily inverted using the Sherman-Morrison formula, see e.g. [124, 19]. In general, however, this exact update cannot be used since multiplication of vectors with $(M-B)^{-1}$ is expensive. Instead, cheap approximations of $(M-B)^{-1}$ must be considered. In the work [121] of Meurant we find approximate updates of incomplete Cholesky factorizations and [14, 20] banded updates were proposed for both symmetric positive definite approximate inverse and incomplete Cholesky preconditioners by Benzi and Bertaccini.

Chapter 4 of this habilitation thesis addresses a more general black-box approximate update scheme for factorized preconditioners which first appeared in $[54]_{DT}$, see also its extensions in $[22]_{DT}$, $[23]_{DT}$, $[55]_{DT}$ and $[56]_{DT}$. The chapter contains the basic paper $[54]_{DT}$, the more applied paper $[23]_{DT}$ with additional theory, generalizations for block decompositions and experiments with problems from computational fluid dynamics, and the paper $[56]_{DT}$ on matrix-free implementations. The proposed preconditioner update is designed for general nonsymmetric linear systems solved by arbitrary iterative solvers and hence it can be combined with some of the techniques for more specific systems and solvers mentioned before. The basic idea is to combine an incomplete reference factorization with a Gauss-Seidel type of approximation of the difference between the current and the reference matrix. The technique tends to correct deteriorating numbers of iterations needed to solve with a frozen preconditioner by reducing them to an acceptable level. Moreover, the updated factorizations can be more powerful than preconditioners computed from scratch; this may happen, for instance, when the updates are related to more stable reference preconditioners generated earlier in the sequence. Since the updated factorizations are both cheap to compute and cheaply applied as a preconditioner, their use is of considerable interest for practice. This is especially true when preconditioner computations are expensive, like in matrix-free environment where the matrix has to be estimated first to be able to construct at all a reference preconditioner. Special techniques like graph coloring have to be used and the application of our updated preconditioner should be carried out in such a way that the difference matrix B is never computed and stored explicitly. These and other issues related to matrix-free implementation are the subject of $[56]_{DT}$. Our work, which was inspired by the work of Benzi and Bertaccini [14, 20], appears to be the first in a series of papers on preconditioner updates for nonsymmetric linear systems [34, 11, 10, 12, 117, 42] and is being referred to in publications from different research areas (see, e.g., [21, 43, 81, 158, 18]).

1.4 Classification with high-dimensional data

The task of classifying an observation to a group based on a number of observations for which the group information is known a priori, forms a classical part of statistics [134, Chapters 8 and 9]. Consider n observations, each represented by a vector $x_i \in \mathbb{R}^p$, $i = 1, \ldots, n$, where p is the number of variables (i.e of observed properties). Assume each observation belongs to one of g groups with $g < n$. A very simple way to assign a new observation $z \in \mathbb{R}^p$ to one of the g groups is to compute its distance to all group means and select the group with closest mean. In other words, z is classified to the kth group, if

$$k = \arg \min_{1 \le j \le g} \|z - \bar{x}_j\|, \tag{1.11}$$

where $\bar{x}_1, \ldots, \bar{x}_g$ are the group means. This way of classification does not take into account potentially different variability of the observed variables. In Linear Discriminant Analysis this is overcome by replacing the Euclidean distance in (1.11) with the Mahalanobis distance. The Mahalanobis distance $\|z - \bar{x}_j\|_M$ is defined with the help of the sample covariance matrix $S \in \mathbb{R}^{p \times p}$, provided it is non-singular, as

$$\|z - \bar{x}_j\|_M \equiv \sqrt{(z - \bar{x}_j)^T S^{-1} (z - \bar{x}_j)}. \tag{1.12}$$

The sample covariance matrix is symmetric positive definite and describes the correlation between the individual variables. Its entry on position (i, j) gives the sample covariance between the ith and jth variable, or, in terms of linear algebra, the inner product between x_i and x_j, both centered through substraction of their sample mean. Using matrix notation, if the given n observations are collected in a matrix $X \in \mathbb{R}^{p \times n}$ whose ith column contains the ith observation and if X is centered by substracting the overall mean denoted $\bar{x}$, then the sample covariance matrix is

$$S = (X - \bar{x}e^T)(X^T - e\bar{x}^T), \tag{1.13}$$

where $e \in \mathbb{R}^n$ is the vector of ones. Yet another way to write this is

$$S = \sum_{i=1}^{n} (x_i - \bar{x}e^T)(x_i^T - e\bar{x}^T). \tag{1.14}$$

We do not include here the usual normalization (division with $n-1$) as it does not influence the outcome of the classification procedure.

When the given data are high-dimensional, i.e. p is large, efficient matrix computations become important for several reasons. First of all, sparsity should be exploited when X is sparse. This is often the case in data mining and pattern recognition applications like, for instance, information retrieval. The observations can, to give an example, represent documents classified according to the presence of certain keywords. As most documents contain only

a few significant keywords (variables), most of the entries of the corresponding term-document matrix X are zero [61, Chapter 1]. If X is sparse, S can be represented with low memory costs (only X and $\bar{x}$ need to be stored, see (1.13)) and a sparse factorization or a Krylov subspace method may be used for the multiplication with S^{-1}.

If p is larger than n (in modern problems often $p \gg n$) an additional issue arises: The covariance matrix will be singular. This is so because its rank is at most n, see (1.14). Then classification based on the Mahalanobis distance (1.12) cannot be carried out. One usually applies some form of regularization to the covariance matrix, for example one adds a small multiple ϵ of the identity matrix [73, 87]. Such techniques do overcome the singularity of S, but from the numerical point of view they are not very efficient. They work with a large, full rank matrix $S^* \in \mathbb{R}^{p \times p}$ (the regularization of S), whereas S is of much smaller rank. Some artificial information has been added, making decompositions for S^* unnecessarily expensive. In addition, regularization requires the time-consuming search for an appropriate regularization parameter (ϵ in the above example).

To reduce the number of variables p, one can perform a preprocessing step named dimension reduction. Principal component analysis (PCA) is a standard technique which uses the singular value decomposition as its main tool, but in the context of classification it is necessary to preserve class information during the dimension reduction. This is achieved by a variant of linear discriminant analysis which can be used for both dimension reduction and classification. *Fisher's* linear discriminant analysis [70] (FLDA) splits the covariance matrix into a matrix B for the variance between groups and a matrix W for the variance within each group. The data are then projected onto a small subspace in which between-group variances are maximized and within-group variances are minimized, thus emphasizing the available group information. These projections are obtained from the solution of the generalized eigenproblem

$$Bv = \lambda W v.$$

If $p < n$, this generalized eigenproblem can be transformed into the standard eigenproblem

$$W^{-1}Bv = \lambda v,$$

or even into a standard symmetric eigenproblem using the Cholesky decomposition of W. But for $p \geq n$, W is singular and similarly as before for the covariance matrix S, one usually applies some form of regularization to W like a small positive shift [97, 38, 106].

In Chapter 5 (i.e., publication $[53]_{DT}$), an FLDA method for $p \gg n$ is proposed which attempts to optimize the involved matrix operations and other numerical aspects. It is not merely a variant of FLDA with a strong numerical linear algebra kernel, but it addresses in detail the best way to cope with singular covariance matrices, both from the classification and the matrix computation point of view. The main idea is to eliminate the common null space

of B and W before starting the FLDA process. The paper shows that this does not affect the performance of FLDA classification, while it takes away precisely the property that can make the solution of generalized eigenproblems numerically most unstable. Moreover, it is shown that elimination of the common null space of B and W amounts to performing a PCA for $X - \bar{x}e$; for a sparse data matrix X this can be done efficiently with an iterative Krylov subspace method for the singular value decomposition, e.g. Golub-Kahan bidiagonalization [79]. For dense data, nearly all computations can be performed in n-dimensional or smaller spaces, making the overall computational costs of the order of pn^2 flops, as opposed to $\mathcal{O}(p^3)$ flops for most regularized versions of FLDA.

A Matlab code $[47]_{DT}$ for our proposed method was incorporated into BIOSIG [1], an open source software library for biomedical signal processing. Major application areas are neuroinformatics, brain-computer interfaces, neurophysiology, psychology, cardiovascular systems and sleep research. A very active research area where the $p > n$ problem arises frequently, is gene expression for cancer diagnostics. Here one classifies genes to disease types based on their gene expressions (i.e., the amino-acids that characterize them), but it is very difficult or even impossible to gather enough genes. Not only is the number of investigated genes in general very small, but in addition the number of variables is significantly larger (for instance thousand times larger).

Recently, the author of this habilitation thesis has tried with collaborators to extend ideas of $[53]_{DT}$ to other variants and aspects of linear discriminant analysis, see e.g. $[100]_{DT}$, $[99]_{DT}$, [5], and, in Czech, [132]. The author of this habilitation thesis has contributed to the publications $[8]_{DT}$, $[91]_{DT}$ and $[147]_{DT}$ through providing various types of statistical analyses.

Bibliography

[1] *BIOSIG open software library.* `http://biosig.cvs.sourceforge.net/`.

[2] P. R. Amestoy, I. S. Duff, and J.-Y. L'Excellent, *Multifrontal parallel distributed symmetric and unsymmetric solvers*, Comput. Methods Appl. Mech. Engrg., (2000), pp. 501–520.

[3] G. S. Ammar and C. Y. He, *On an inverse eigenvalue problem for unitary Hessenberg matrices*, Linear Algebra Appl., 218 (1995), pp. 263–271.

[4] M. Arioli, V. Pták, and Z. Strakoš, *Krylov sequences of maximal length and convergence of GMRES*, BIT, 38 (1998), pp. 636–643.

[5] S. Athanasiadis, *The small sample size problem in gene expression tasks*, Diploma thesis, Charles University, 2015.

[6] C. Beattie, M. Embree, and J. Rossi, *Convergence of restarted Krylov subspaces to invariant subspaces*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 1074–1109.

[7] C. A. Beattie, M. Embree, and D. C. Sorensen, *Convergence of polynomial restart Krylov methods for eigenvalue computations*, SIAM Rev., 47 (2005), pp. 492–515.

[8] T. Belkina, D. Khojiev, M. Tillyashaykhov, Z. Tigay, M. Kudenov, J. Duintjer Tebbens, and J. Vček, *Delay in the diagnosis and treatment of pulmonary tuberculosis in uzbekistan: A cross-sectional study*, BMC Infectious Diseases, 14 (2014).

[9] M. Bellalij and H. Sadok, *A new approach to GMRES convergence*, unpublished report, 2011.

[10] S. Bellavia, D. Bertaccini, and B. Morini, *Nonsymmetric preconditioner updates in Newton-Krylov methods for nonlinear systems*, SIAM J. Sci. Comput., 33 (2011), pp. 2595–2619.

[11] S. Bellavia, V. De Simone, D. di Serafino, and B. Morini, *Efficient preconditioner updates for shifted linear systems*, SIAM J. Sci. Comput., 33 (2011), pp. 1785–1809.

[12] S. Bellavia, B. Morini, and M. Porcelli, *New updates of incomplete LU factorizations and applications to large nonlinear systems*, Optim. Methods Softw., 29 (2014), pp. 321–340.

[13] M. Benzi, *Preconditioning techniques for large linear systems: a survey*, J. Comput. Phys., 182 (2002), pp. 418–477.

[14] M. Benzi and D. Bertaccini, *Approximate inverse preconditioning for shifted linear systems*, 43 (2003), pp. 231–244.

[15] M. Benzi, J. Haws, and M. Tůma, *Preconditioning highly indefinite and nonsymmetric matrices*, SIAM J. Sci. Comput., 21 (2000), pp. 1333–1353.

[16] M. Benzi, C. D. Meyer, and M. Tůma, *A sparse approximate inverse preconditioner for the conjugate gradient method*, SIAM J. Sci. Comput., 17 (1996), pp. 1135–1149.

[17] M. Benzi and M. Tůma, *A sparse approximate inverse preconditioner for nonsymmetric linear systems*, SIAM J. Sci. Comput., 19 (1998), pp. 968–994.

[18] L. Berenguer and D. Tromeur-Dervout, *Asynchronous partial update of the restricted additive Schwarz preconditioner to solve nonlinear CFD problems*, Comput. & Fluids, 110 (2015), pp. 211–218.

[19] L. Bergamaschi, R. Bru, A. Martínez, and M. Putti, *Quasi-Newton preconditioners for the inexact Newton method*, Electron. Trans. Numer. Anal., 23 (2006), pp. 76–87.

[20] D. Bertaccini, *Efficient preconditioning for sequences of parametric complex symmetric linear systems*, Electron. Trans. Numer. Anal., 18 (2004), pp. 49–64.

[21] D. Bertaccini and F. Sgallari, *Updating preconditioners for nonlinear deblurring and denoising image restoration*, Appl. Numer. Math., 60 (2010), pp. 994–1006.

[22] P. Birken, J. Duintjer Tebbens, A. Meister, and M. Tůma, *Updating preconditioners for permuted nonsymmetric linear systems*, Proc. Appl. Math. Mech., 7 (2007), pp. 1022101–1022102.

[23] ——, *Preconditioner updates applied to CFD model problems*, Appl. Num. Math., 58 (2008), pp. 1628–1641.

[24] C. H. Bischof, *Incremental condition estimation*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 312–322.

[25] C. H. Bischof, J. G. Lewis, and D. J. Pierce, *Incremental condition estimation for sparse matrices*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 644–659.

[26] C. H. Bischof and P. T. P. Tang, *Generalizing incremental condition estimation*, J. Numer. Linear Algebra Appl., 1 (1992), pp. 149–163.

[27] M. Bollhöfer, *A robust ILU with pivoting based on monitoring the growth of the inverse factors*, Lin. Alg. Appl., 338 (2001), pp. 201–218.

[28] ——, *A robust and efficient ILU that incorporates the growth of the inverse triangular factors*, SIAM J. Sci. Comput., 25 (2003), pp. 86–103.

[29] ——, *ILUPACK: version 2.4*, 2011. `www.icm.tu-bs.de/~bolle/ilupack/`.

[30] R. P. Brent, *Some efficient algorithms for solving systems of nonlinear equations*, SIAM J. Numer. Anal., 10 (1973), pp. 327–344.

[31] R. Bru, J. Cerdán, J. Marín, and J. Mas, *Preconditioning sparse nonsymmetric linear systems with the Sherman-Morrison formula*, SIAM J. Sci. Comput., 25 (2003), pp. 701–715.

[32] R. Bru, J. Marín, J. Mas, and M. Tůma, *Balanced incomplete factorization*, SIAM J. Sci. Comput., 30 (2008), pp. 2302–2318.

[33] ——, *Improved balanced incomplete factorization*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2431–2452.

[34] C. Calgaro, J.-P. Chehab, and Y. Saad, *Incremental incomplete LU factorizations with applications*, Numer. Linear Algebra Appl., 17 (2010), pp. 811–837.

[35] B. Carpentieri, I. S. Duff, and L. Giraud, *A class of spectral two-level preconditioners*, SIAM J. Sci. Comput., 25 (2003), pp. 749–765.

[36] A. Chapman and Y. Saad, *Deflated and augmented Krylov subspace techniques*, Numer. Linear Algebra Appl., 4 (1997), pp. 43–66.

[37] K. Chen, *Matrix preconditioning techniques and applications*, vol. 19 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2005.

[38] Z. Y.-M. Cheng, Y.-Q. and J.-Y. Yang, *Optimal fisher discriminant analysis using the rank decomposition*, Pattern Recognition, 25 (1992), pp. 101–111.

[39] E. Chow and Y. Saad, *Experimental study of ILU preconditioners for indefinite matrices*, J. Comput. Appl. Math., 86 (1997), pp. 387–414.

[40] T. A. Davis, *Direct methods for sparse linear systems*, vol. 2 of Fundamentals of Algorithms, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2006.

[41] T. A. Davis and I. S. Duff, *An unsymmetric-pattern multifrontal method for sparse LU factorization*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 140–158.

[42] V. De Simone and D. di Serafino, *A matrix-free approach to build band preconditioners for large-scale bound-constrained optimization*, J. Comput. Appl. Math., 268 (2014), pp. 82–92.

[43] V. Dolejší, M. Holík, and J. Hozman, *Efficient solution strategy for the semi-implicit discontinuous Galerkin discretization of the Navier-Stokes equations*, J. Comput. Phys., 230 (2011), pp. 4176–4200.

[44] I. S. Duff and C. Vömel, *Incremental norm estimation for dense and sparse matrices*, BIT, 42 (2002), pp. 300–322.

[45] J. Duintjer Tebbens, *An application of the sherman-morrison formula to the gmres method*, in ,,Conjugate Gradient Algorithms and Finite Element Methods", Springer-Verlag, Berlin, (Kek, M.; Neittaanmki, P.; Glowinski, R.; Korotov, S.), (2004), pp. 69–92.

[46] ——, *Modern methods for solving linear systems*, PhD thesis, Faculty of Mathematics and Physics, Charles University, Prague, 2004.

[47] ——, *Sparse LDA*, 2007. `http://biosig.cvs.sourceforge.net/biosig/biosig/t400/train_lda_sparse.m?view=markup`.

[48] J. Duintjer Tebbens, I. Hnětynková, M. Plešinger, Z. Strakoš, and P. Tichý, *Analýza metod pro maticové výpočty – Základní metody*, První vydání, Matfyzpress, Praha, 2012.

[49] J. Duintjer Tebbens and G. Meurant, *On the convergence of Q-OR and Q-MR Krylov methods for solving nonsymmetric linear systems*, BIT, to appear.

[50] ——, *Any Ritz value behavior is possible for Arnoldi and for GMRES*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 958–978.

[51] ——, *Prescribing the behavior of early terminating GMRES and Arnoldi iterations*, Num. Algor., 65 (2014), pp. 69–90.

[52] J. Duintjer Tebbens, G. Meurant, H. Sadok, and Z. Strakoš, *On investigating GMRES convergence using unitary matrices*, Lin. Alg. Appl., 450 (2014), pp. 83–107.

[53] J. Duintjer Tebbens and P. Schlesinger, *Improving implementation of linear discriminant analysis for the high dimension/small sample size problem*, Comput. Statist. Data Anal., 52 (2007), pp. 423–437.

[54] J. Duintjer Tebbens and M. Tůma, *Efficient preconditioning of sequences of nonsymmetric linear systems*, SIAM J. Sci. Comput., 29 (2007), pp. 1918–1941.

[55] J. Duintjer Tebbens and M. Tůma, *Improving triangular preconditioner updates for nonsymmetric linear systems*, in Large-scale scientific computing, vol. 4818 of Lecture Notes in Comput. Sci., Springer, Berlin, 2008, pp. 737–744.

[56] ——, *Preconditioner updates for solving sequences of linear systems in matrix-free environment*, Numer. Linear Algebra Appl., 17 (2010), pp. 997–1019.

[57] J. Duintjer Tebbens and M. Tůma, *On incremental condition estimators in the 2-norm*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 174–197.

[58] M. Eiermann, *Fields of values and iterative methods*, Linear Algebra Appl., 180 (1993), pp. 167–197.

[59] M. Eiermann and O. G. Ernst, *Geometric aspects of the theory of Krylov subspace methods*, Acta Numer., 10 (2001), pp. 251–312.

[60] S. Eisenstat and H. Walker, *Choosing the forcing terms in an inexact Newton method*, SIAM J. Sci. Comput., 17 (1996), pp. 16–32.

[61] L. Eldén, *Matrix methods in data mining and pattern recognition*, vol. 4 of Fundamentals of Algorithms, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2007.

[62] H. C. Elman and A. Ramage, *Fourier analysis of multigrid for a model two-dimensional convection-diffusion equation*, BIT, 46 (2006), pp. 283–306.

[63] M. Embree, *The Arnoldi eigenvalue iteration with exact shifts can fail*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 1–10.

[64] J. Erhel, K. Burrage, and B. Pohl, *Restarted GMRES preconditioned by deflation*, J. Comput. Appl. Math., 69 (1996), pp. 303–318.

[65] T. Ericsson, *On the eigenvalues and eigenvectors of Hessenberg matrices*, Numerical Analysis Group,Report 10, Chalmers University of Technology and the University of Göteborg,Sweden, available online at http://www.cs.chalmers.se/pub/num_analysis/reports/, June 1990.

[66] V. Faber, J. Liesen, and P. Tichý, *The Faber-Manteuffel theorem for linear operators*, SIAM J. Numer. Anal., 46 (2008), pp. 1323–1337.

[67] V. Faber and T. Manteuffel, *Necessary and sufficient conditions for the existence of a conjugate gradient method*, SIAM J. Numer. Anal., 21 (1984), pp. 352–362.

[68] K. Fan and G. Pall, *Imbedding conditions for Hermitian and normal matrices*, Canad. J. Math., 9 (1957), pp. 298–304.

[69] M. Fiedler, *Special matrices and their applications in numerical mathematics*, Dover Publications, Inc., Mineola, NY, second ed., 2008.

[70] R. A. Fisher, *The use of multiple measurements in taxonomic problems*, Annals of Eugenics, 7 (1936), pp. 179–188.

[71] R. Fletcher, *Conjugate gradient methods for indefinite systems*, in Numerical analysis (Proc 6th Biennial Dundee Conf., Univ. Dundee, Dundee, 1975). Lecture Notes in Math., Vol. 506, Springer, Berlin, 1976, pp. 73–89.

[72] R. W. Freund and N. M. Nachtigal, *QMR: a quasi-minimal residual method for non-Hermitian linear systems*, Numer. Math., 60 (1991), pp. 315–339.

[73] J. S. Friedman, *Regularized discriminant analysis*, Journal of the American Statistical Assocation, 48 (1989), pp. 165–175.

[74] A. Frommer and U. Glässner, *Restarted GMRES for shifted linear systems*, SIAM J. Sci. Comput., 19 (1998), pp. 15–26.

[75] J. R. Gilbert and T. Peierls, *Sparse partial pivoting in time proportional to arithmetic operations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 862–874.

[76] J. R. Gilbert and S. Toledo, *High-performance out-of-core sparse LU factorization*, in Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing 1999 (San Antonio, TX), SIAM, Philadelphia, PA, 1999, 10 pp.

[77] L. Giraud, S. Gratton, and E. Martin, *Incremental spectral preconditioners for sequences of linear systems*, ANM, 57 (2007), pp. 1164–1180.

[78] L. Giraud, S. Gratton, X. Pinel, and X. Vasseur, *Flexible GMRES with deflated restarting*, SIAM J. Sci. Comput., 32 (2010), pp. 1858–1878.

[79] G. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, J. Soc. Indust. Appl. Math. Ser. B Numer. Anal., 2 (1965), pp. 205–224.

[80] G. H. Golub and C. F. Van Loan, *Matrix computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, third ed., 1996.

[81] S. González-Pintor, D. Ginestar, and G. Verdú, *Preconditioning the solution of the time-dependent neutron diffusion equation by recycling Krylov subspaces*, Int. J. Comput. Math., 91 (2014), pp. 42–52.

[82] A. Greenbaum, *Iterative methods for solving linear systems*, SIAM, Philadelphia, 1997.

[83] A. Greenbaum, *Generalizations of the field of values useful in the study of polynomial functions of a matrix*, Linear Algebra Appl., 347 (2002), pp. 233–249.

[84] A. Greenbaum, V. Pták, and Z. Strakoš, *Any nonincreasing convergence curve is possible for GMRES*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 465–469.

[85] A. Greenbaum and Z. Strakoš, *Matrices that generate the same Krylov residual spaces*, in Recent advances in iterative methods, vol. 60 of IMA Vol. Math. Appl., Springer, New York, 1994, pp. 95–118.

[86] M. J. Grote and T. Huckle, *Parallel preconditioning with sparse approximate inverses*, SIAM J. Sci. Comput., 18 (1997), pp. 838–853.

[87] Y. Guo, T. Hastie, and R. Tibshirani, *Regularized discriminant analysis and its application in microarrays*, Biostatistics, 8 (2007), pp. 86–100.

[88] A. Gupta, *Highly scalable parallel algorithms for sparse matrix factorization*, IEEE Trans. Parallel Distrib. Systems, (1997), pp. 502–520.

[89] W. W. Hager, *Condition estimates*, J Sci. Stat. Comput., 5 (1984), pp. 311–316.

[90] S. Hartmann, J. Duintjer Tebbens, K. J. Quint, and A. Meister, *Iterative solvers within sequences of large linear systems in non-linear structural mechanics*, ZAMM Z. Angew. Math. Mech., 89 (2009), pp. 711–728.

[91] T. Hendrychova, M. Vytrisalova, A. Alwarafi, J. Duintjer Tebbens, H. Vankatova, S. Leal, A. Kubena, A. Smahelova, and J. Vlček, *Fat-related and fiber-related diet behavior among type 2 diabetes patients from distinct regions*, Patient Preference and Adherence, 9 (2014), pp. 319–325.

[92] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp. 409–436 (1953).

[93] N. J. Higham, *A survey of condition number estimation for triangular matrices*, SIAM Rev., 29 (1987), pp. 575–596.

[94] ——, *FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation*, ACM Trans. Math. Softw., 14 (1988), pp. 381–396 (1989).

[95] ——, *Experience with a matrix norm estimator*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 804–809.

[96] N. J. Higham and F. Tisseur, *A block algorithm for matrix 1-norm estimation, with an application to 1-norm pseudospectra*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1185–1201.

[97] Z.-Q. Hong and J.-Y. Yang, *Optimal discriminant plane for a small number of samples and design method of classifier on the plane*, Pattern Recognition, 24 (1991), pp. 317–324.

[98] M. Huhtanen and O. Nevanlinna, *Minimal decompositions and iterative methods*, Numer. Math., 86 (2000), pp. 257–281.

[99] J. Kalina and J. Duintjer Tebbens, *Algorithms for regularized linear discriminant analysis*, Proceedings of BIOINFORMATICS 2015, Scitepress, Lissabon, (2015), pp. 128–133.

[100] J. Kalina, Z. Valenta, and J. Duintjer Tebbens, *Computation of regularized linear discriminant analysis*, Proceedings of COMPSTAT, M. Gilli, G. Gonzalez-Rodrigues, A. Nieto-Reyes (Eds.), Universite de Geneve, (2014), pp. 1–8.

[101] C. Kelley, *Iterative methods for linear and nonlinear equations*, SIAM, Philadelphia, 1995.

[102] ——, *Solving nonlinear equations with Newton's method*, SIAM, Philadelphia, 2003.

[103] S. A. Kharchenko and A. Y. Yeremin, *Eigenvalue translation based preconditioners for the* GMRES(k) *method*, Numer. Linear Algebra Appl., 2 (1995), pp. 51–77.

[104] D. A. Knoll and D. E. Keyes, *Jacobian-free Newton-Krylov methods: a survey of approaches and applications*, J. Comput. Phys., 193 (2004), pp. 357–397.

[105] A. N. Krylov, *On the numerical solution of the equation by which the frequency of small oscillations is determined in technical problems*, Izv. Akad. Nauk SSSR Ser. Fiz.-Mat., 4 (1931), pp. 491–539.

[106] J. P. M.-W. Krzanowski, W.J. and M. Thomas, *Discriminant analysis with singular covariance matrices: Methods and applications to spectroscopic data*, Applied Statistics, 44 (1995), pp. 101–115.

[107] A. B. J. Kuijlaars, *Convergence analysis of Krylov subspace iterations with methods from potential theory*, SIAM Rev., 48 (2006), pp. 3–40.

[108] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Research Nat. Bur. Standards, 45 (1950), pp. 255–282.

[109] ——, *Solution of systems of linear equations by minimized iterations*, J. Research Nat. Bur. Standards, 49 (1952), pp. 33–53.

[110] J. Liesen, *Construction and analysis of polynomial iterative methods for non-Hermitian systems of linear equations*, PhD thesis, University of Bielefeld, Germany, 1998.

[111] ——, *Computable convergence bounds for GMRES*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 882–903.

[112] J. Liesen and Z. Strakoš, *On optimal short recurrences for generating orthogonal Krylov subspace bases*, SIAM Rev., 50 (2008), pp. 485–503.

[113] J. Liesen and Z. Strakoš, *Krylov subspace methods, Principles and analysis*, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, 2013.

[114] J. Liesen and P. Tichý, *Convergence analysis of Krylov subspace methods*, GAMM Mitt. Ges. Angew. Math. Mech., 27 (2004), pp. 153–173 (2005).

[115] ——, *The worst-case GMRES for normal matrices*, BIT, 44 (2004), pp. 79–98.

[116] D. Loghin, D. Ruiz, and A. Touhami, *Adaptive preconditioners for nonlinear systems of equations*, J. Comput. Appl. Math., 189 (2006), pp. 362–374.

[117] W.-H. Luo, T.-Z. Huang, L. Li, Y. Zhang, and X.-M. Gu, *Efficient preconditioner updates for unsymmetric shifted linear systems*, Comput. Math. Appl., 67 (2014), pp. 1643–1655.

[118] S. M. Malamud, *Inverse spectral problem for normal matrices and the Gauss-Lucas theorem*, Trans. Amer. Math. Soc., 357 (2005), pp. 4043–4064.

[119] Mathworks, Inc., *MATLAB 8.5*, 2015. `http://www.mathworks.com/products/matlab/.`

[120] G. Meurant, *Computer solution of large linear systems*, vol. 28 of Studies in Mathematics and its Applications, North-Holland Publishing Co., Amsterdam, 1999.

[121] ——, *On the incomplete Cholesky decomposition of a class of perturbed matrices*, SIAM J. Sci. Comput., 23 (2001), pp. 419–429.

[122] G. Meurant and J. Duintjer Tebbens, *The role eigenvalues play in forming GMRES residual norms with non-normal matrices*, Numerical Algorithms, 68 (2015), pp. 143–165.

[123] G. Meurant and Z. Strakoš, *The Lanczos and conjugate gradient algorithms in finite precision arithmetic*, Acta Numerica, 15 (2006), pp. 471–542.

[124] J. Morales and J. Nocedal, *Automatic preconditioning by limited-memory quasi-Newton updates*, SIAM J. Opt., 10 (2000), pp. 1079–1096.

[125] R. B. Morgan, *A restarted GMRES method augmented with eigenvectors*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1154–1171.

[126] N. M. Nachtigal, S. C. Reddy, and L. N. Trefethen, *How fast are nonsymmetric matrix iterations?*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 778–795.

[127] C. C. Paige, *The computation of eigenvalues and eigenvectors of very large sparse matrices*, PhD thesis, University of London, 1971.

[128] ——, *Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix*, J. Inst. Math. Appl., 18 (1976), pp. 341–349.

[129] ——, *Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem*, Linear Algebra Appl., 34 (1980), pp. 235–258.

[130] C. C. Paige and M. A. Saunders, *Solutions of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.

[131] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti, *Recycling Krylov subspaces for sequences of linear systems*, SIAM J. Sci. Comput., 28 (2006), pp. 1651–1674.

[132] V. Pekař, *Efektivní implementace metod pro redukci dimenze v mnohorozměrné statistice*, Diploma thesis, Charles University, Prague, 2015.

[133] J. Pestana and A. Wathen, *On choice of preconditioner for minimum residual methods for non-hermitian matrices*, J. Comput. Appl. Math., 249 (2013), pp. 57–68.

[134] A. C. Rencher, *Methods of multivariate analysis*, Wiley Series in Probability and Statistics, Wiley-Interscience, New York, second ed., 2002.

[135] Y. Saad, *Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices*, Linear Algebra Appl., 34 (1980), pp. 269–295.

[136] ——, *Krylov subspace methods for solving large unsymmetric linear systems*, Math. Comp., 37 (1981), pp. 105–126.

[137] ——, *The Lanczos biorthogonalization algorithm and other oblique projection methods for solving large unsymmetric systems*, SIAM J. Numer. Anal., 19 (1982), pp. 485–506.

[138] Y. Saad, *Numerical methods for large eigenvalue problems*, Algorithms and Architectures for Advanced Scientific Computing, Manchester University Press, Manchester; Halsted Press [John Wiley & Sons, Inc.], New York, 1992.

[139] Y. Saad, *Iterative methods for sparse linear systems*, PWS Publishing Co., Boston, 1996.

[140] Y. Saad and M. H. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[141] O. Schenk, K. Gärtner, and W. Fichtner, *Efficient sparse LU factorization with left-right looking strategy on shared memory multiprocessors*, BIT, 40 (2000), pp. 158–176.

[142] V. V. Shaĭdurov, *Multigrid methods for finite elements*, vol. 318 of Mathematics and its Applications, Kluwer Academic Publishers Group, 1995.

[143] V. Shamanskii, *A modification of Newton's method*, Ukrain. Mat. Z., 19 (1967), pp. 1333–1338.

[144] N. Shomron and B. N. Parlett, *Linear algebra meets Lie algebra: the Kostant-Wallach theory*, Linear Algebra Appl., 431 (2009), pp. 1745–1767.

[145] G. Shroff and C. Bischof, *Adaptive condition estimation for rank-one updates of QR factorizations*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1264–1278.

[146] V. Simoncini and E. Gallopoulos, *An iterative method for nonsymmeric systems with multiple right-hand sides*, SIAM J. Sci. Comput., 16 (1995), pp. 917–933.

[147] T. Smutný, J. Duintjer Tebbens, and P. Pávek, *Bioinformatic analysis of mirnas targeting the key nuclear receptors regulating cyp3a4 gene expression: The challenge of the cyp3a4 "missing heritability" enigma*, Journal of Applied Biomedicine, 13 (2015), pp. 181–188.

[148] P. Stange, A. Griewank, and M. Bollhöfer, *On the efficient update of rectangular LU factorizations subject to low rank modifications*, ETNA, 26 (2007), pp. 161–177.

[149] Z. Strakoš and J. Liesen, *On numerical stability in large scale linear algebraic computations*, ZAMM Z. Angew. Math. Mech., 85 (2005), pp. 307–325.

[150] U. H. Suhl and L. M. Suhl, *Computing sparse LU factorizations for large-scale linear programming bases*, ORSA J. Computing, 2 (1990), pp. 325–335.

[151] L. N. Trefethen and M. Embree, *Spectra and pseudospectra*, Princeton University Press, Princeton, NJ, 2005.

[152] A. van der Sluis and H. A. van der Vorst, *The rate of convergence of conjugate gradients*, Numer. Math., 48 (1986), pp. 543–560.

[153] H. A. van der Vorst, *Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.

[154] H. A. van der Vorst, *Iterative Krylov methods for large linear systems*, vol. 13 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2009.

[155] H. A. van der Vorst and C. Vuik, *The superlinear convergence behaviour of GMRES*, J. Comput. Appl. Math., 48 (1993), pp. 327–341.

[156] R. S. Varga, *Matrix iterative analysis*, vol. 27 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, expanded ed., 2000.

[157] E. Vecharynski and J. Langou, *Any admissible cycle-convergence behavior is possible for restarted GMRES at its initial cycles*, Num. Lin. Algebr. Appl., 18 (2011), pp. 499–511.

[158] W. Xu and T. F. Coleman, *Solving nonlinear equations with the Newton-Krylov method based on automatic differentiation*, Optim. Methods Softw., 29 (2014), pp. 88–101.

SIAM J. MATRIX ANAL. APPL.
Vol. 33, No. 3, pp. 958–978

ANY RITZ VALUE BEHAVIOR IS POSSIBLE FOR ARNOLDI AND FOR GMRES*

JURJEN DUINTJER TEBBENS† AND GÉRARD MEURANT‡

Abstract. We show that arbitrary convergence behavior of Ritz values is possible in the Arnoldi method, and we give two parametrizations of the class of matrices with initial Arnoldi vectors that generate prescribed Ritz values (in all iterations). The second parametrization enables us to prove that any GMRES residual norm history is possible with any prescribed Ritz values (in all iterations), provided that we treat the stagnation case appropriately.

Key words. Ritz values, Arnoldi process, Arnoldi method, GMRES method, prescribed convergence, interlacing properties

AMS subject classifications. 65F15, 65F10, 65F18, 15A18

DOI. 10.1137/110843666

1. Introduction.

Let A be a nonsingular matrix of order n and b a nonzero n-dimensional vector. The Arnoldi process [3] reduces A to upper Hessenberg form by a particular type of Gram–Schmidt orthogonalization for the vectors $b, Ab, A^2b, \ldots$. At each step of the process, one matrix-vector multiplication with A is performed, and one row and one column are appended to the previous Hessenberg matrix. The process is well suited to iterative methods with large sparse matrices A. Two popular methods for extracting approximate solutions from the generated Hessenberg matrices are the generalized minimal residual (GMRES) method [40] for solving the linear system $Ax = b$ and the Arnoldi method (see, e.g., [38, 39]) for computing the eigenvalues and eigenvectors of A.

The Arnoldi process can be seen as a generalization to non-Hermitian matrices of the Lanczos process for tridiagonalization of Hermitian matrices [24]. The Lanczos process is at the basis of the conjugate gradients (CG) method [23, 25] for Hermitian positive definite linear systems and of the Lanczos method for Hermitian eigenproblems [24]. In this sense GMRES is a generalization of CG (even though the l_2 norm of the residual is not minimized in CG), and the Arnoldi method is a generalization of the Lanczos method. As convergence of the CG and Lanczos methods are well understood, it is natural to take the convergence theory of these methods as a starting point for explaining the behavior of the GMRES and Arnoldi methods. In the CG method, the convergence behavior is dictated by the distribution of the eigenvalues of the matrix. In practice, the same is often observed for the GMRES method, but, with possibly nonnormal input matrices, the situation becomes more subtle. For example, Greenbaum and Strakoš [22] proved that if a residual norm convergence

*Received by the editors August 8, 2011; accepted for publication (in revised form) by Q. Ye May 3, 2012; published electronically September 5, 2012.
http://www.siam.org/journals/simax/33-3/84366.html

†Institute of Computer Science, Academy of Sciences of the Czech Republic, 18 207 Praha 8-Libeň, Czech Republic (duintjertebbens@cs.cas.cz). This author's work is part of the Institutional Research Plan AV0Z10300504 and was supported by project IAA100300802 of the Grant Agency of the ASCR and by project M100300901 of the institutional support of the ASCR.

‡30 rue du sergent Bauchat, 75012 Paris, France (gerard.meurant@gmail.com). The work on this paper was started in 2010 during this author's visit to the Nečas Center of Charles University in Prague supported by a grant from the Jindrich Nečas Center for Mathematical Modeling, project LC06052, financed by MSMT.

curve is generated by GMRES, the same curve can be obtained with a matrix having prescribed nonzero eigenvalues (see [12, Lemma 6.9] for an analogue on prescribed nonzero singular values). Greenbaum, Pták, and Strakoš [21] complemented their result by proving that *any* nonincreasing sequence of residual norms can be given by GMRES (a similar result for residual norms at the end of restart cycles in the restarted GMRES method can be found in [47]). Furthermore, in Arioli, Pták, and Strakoš [2] a complete parametrization was given of all pairs $\{A, b\}$ generating a prescribed residual norm convergence curve and such that A has a prescribed spectrum. The results in these papers show that the GMRES residual norm convergence need not, in general, depend on the eigenvalues of A alone. Other objects, mostly closely related to eigenvalues, have been considered to explain convergence, for example, the pseudospectrum [44], the field of values [11], or the numerical polynomial hull [20]. In [46] it was suggested that convergence of the eigenvalues of the Hessenberg matrices generated in the Arnoldi process (the so-called Ritz values) to eigenvalues of A will often explain the acceleration of convergence of GMRES.

A fundamental tool in the convergence analysis of the Lanczos method for Hermitian eigenproblems is the interlacing property for the eigenvalues of the subsequently generated tridiagonal matrices. It enables one to prove, among other things, the persistence theorem on stabilization of Ritz values (see, e.g., [32, 33, 34] or [31]). There are several generalizations of the interlacing property to normal matrices; see, e.g., [16, 1] or the papers [27, 14] with geometric interpretations. However, just as for GMRES, potentially nonnormal input matrices make convergence analysis of the Arnoldi method delicate. There is no interlacing property for the principal submatrices of general nonnormal matrices; see [42] for a thorough discussion on this topic and its relation to the field of Lie algebra. In [9, 10] one finds a sufficient and necessary condition for prescribing arbitrary eigenvalues of (not necessarily principal) submatrices of general non-Hermitian matrices. For a detailed spectral analysis of nonnormal Hessenberg matrices and their principal submatrices, see also [49].

Since the GMRES and the Arnoldi methods are closely related through the Arnoldi orthogonalization process, a naturally arising question is whether a result, similar to the results of Arioli, Greenbaum, Pták, and Strakoš, on arbitrary convergence behavior of the Arnoldi method can be proved. By arbitrary convergence behavior of the Arnoldi method, we mean the ability to prescribe *all* Ritz values from the very first until the very last iteration (we do not consider convergence to eigenvectors). In this paper we will give a parametrization of the class of all matrices and initial Arnoldi vectors that generates prescribed Ritz values. Besides this result on arbitrary convergence behavior of the Arnoldi method, we derive a parametrization that allows us to characterize all pairs $\{A, b\}$ generating arbitrary convergence behavior of *both* GMRES and Arnoldi. The Ritz values generated in the GMRES method therefore do not, in general, have any influence on the generated residual norms.

The paper is organized as follows: In the remainder of this section we introduce some notation, in particular the notation used in [2], which we adopt, and we recall the parametrization given in [2]. In section 2 we give a parametrization of the class of matrices and initial Arnoldi vectors that generates prescribed Ritz values. Section 3 reformulates the parametrization in order to parametrize the pairs $\{A, b\}$ generating arbitrary behavior of GMRES and Arnoldi at the same time. We close with a brief discussion of our results and some words on future work.

1.1. Notation. We will use the following parametrization of matrices and right-hand sides giving prescribed spectrum and prescribed convergence of the GMRES method (see Theorem 2.1 and Corollary 2.4 of [2]).

THEOREM 1.1. *Assume that we are given n positive numbers*

$$f(0) \geq f(1) \geq \cdots \geq f(n-1) > 0$$

and n complex numbers $\lambda_1, \ldots, \lambda_n$ all different from 0. Let A be a matrix of order n and b an n-dimensional vector. The following assertions are equivalent:

1. *The spectrum of A is $\{\lambda_1, \ldots, \lambda_n\}$, and GMRES applied to A and b with zero initial guess yields residuals $r^{(k)}$, $k = 0, \ldots, n-1$, such that*

$$\|r^{(k)}\| = f(k), \quad k = 0, \ldots, n-1.$$

2. *The matrix A is of the form*

$$A = WYC^{(n)}Y^{-1}W^*$$

and $b = Wh$, where W is a unitary matrix; Y is given by

$$Y = \begin{bmatrix} h & R \\ 0 & \end{bmatrix}, \tag{1.1}$$

with R being a nonsingular upper triangular matrix of order $n-1$ and h a vector such that

$$h = [\eta_1, \ldots, \eta_n]^T, \qquad \eta_k = (f(k-1)^2 - f(k)^2)^{1/2}, \quad k < n, \quad \eta_n = f(n-1); \tag{1.2}$$

and $C^{(n)}$ is the companion matrix corresponding to the polynomial $q(\lambda)$ defined as

$$q(\lambda) = (\lambda - \lambda_1) \cdots (\lambda - \lambda_n) = \lambda^n + \sum_{j=0}^{n-1} \alpha_j \lambda^j,$$

$$C^{(n)} = \begin{bmatrix} 0 & & -\alpha_0 \\ & \ddots & \vdots \\ & I_{n-1} & -\alpha_{n-1} \end{bmatrix}.$$

Furthermore, we will denote by e_j the jth column of the identity matrix of appropriate order. For a matrix M, the leading principal submatrix of order k will be denoted by M_k. With "the subdiagonal" and "subdiagonal entries" we mean the (entries on the) first diagonal under the main diagonal. Throughout the paper we assume exact arithmetics, and we also assume that the investigated Arnoldi processes do not terminate before the nth iteration. This means that the input matrix must be nonderogatory. Note that Theorem 1.1 assumes this situation. The case of early termination will be treated in a forthcoming paper.

2. Prescribed convergence of Ritz values in Arnoldi's method. Consider the kth iteration of an Arnoldi process with a matrix A and initial vector b where an upper Hessenberg matrix H_k (with entries $h_{i,j}$) is generated satisfying

$$AV^{(k)} = V^{(k)}H_k + h_{k+1,k}\, v_{k+1} e_k^T, \quad k < n, \tag{2.1}$$

with ${V^{(k)}}^* V^{(k)} = I_k$, $V^{(k)} e_1 = b/\|b\|$, and ${V^{(k)}}^* v_{k+1} = 0$, $V^{(k)}$ being the matrix whose columns are the basis vectors $v_1, \ldots, v_k$ of the kth Krylov subspace $\mathcal{K}_k(A, b) \equiv \text{span}\{b, Ab, \ldots, A^{k-1}b\}$. The eigenvalues of H_k give the k-tuple

$$\mathcal{R}^{(k)} = (\rho_1^{(k)}, \ldots, \rho_k^{(k)})$$

of the k (not necessarily distinct) Ritz values generated at the kth iteration of Arnoldi's method. We denote by $\mathcal{R}$ the set

$$\mathcal{R} \equiv \{\mathcal{R}^{(1)}, \mathcal{R}^{(2)}, \ldots, \mathcal{R}^{(n)}\}$$

representing all $(n+1)n/2$ generated Ritz values. We also use $\mathcal{S}$ for the *strict* Ritz values without the spectrum of A, i.e.,

$$\mathcal{S} \equiv \mathcal{R} \setminus \mathcal{R}^{(n)},$$

and we will denote the (not necessarily distinct) eigenvalues of the input matrix by $\lambda_1, \ldots, \lambda_n$, i.e.,

$$\mathcal{R}^{(n)} = (\lambda_1, \ldots, \lambda_n).$$

In this section we investigate whether the Arnoldi method can generate arbitrary Ritz values in all iterations. The Ritz values in the Arnoldi method are eigenvalues of the leading principal submatrices of upper Hessenberg matrices with positive real subdiagonal entries. Prescribing the set $\mathcal{R}$ is possible only if there exist, at all, Hessenberg matrices with positive subdiagonal entries where the eigenvalues of all the leading principal submatrices can be prescribed. Parlett and Strang proved that there is a unique upper Hessenberg matrix with the entry one along the subdiagonal such that all leading principal submatrices have arbitrary prescribed eigenvalues; see [36, Theorem 3]. We give here a characterization of this unique matrix, which we denote with $H(\mathcal{R})$, that shows how it is constructed from the prescribed Ritz values.

PROPOSITION 2.1. *Let the set*

$$\begin{aligned}\mathcal{R} = \{\, & \rho_1^{(1)}, \\ & (\rho_1^{(2)}, \rho_2^{(2)}), \\ & \vdots \\ & (\rho_1^{(n-1)}, \ldots, \rho_{n-1}^{(n-1)}), \\ & (\lambda_1, \ldots\ldots\ldots, \lambda_n)\,\}\end{aligned}$$

represent any choice of $n(n+1)/2$ complex Ritz values, and denote the $k \times k$ companion matrix of the polynomial with roots $\rho_1^{(k)}, \ldots, \rho_k^{(k)}$ by $C^{(k)}$. If we define the kth column of the unit upper triangular matrix $U(\mathcal{S})$ through

$$U(\mathcal{S})\, e_1 = e_1, \qquad U(\mathcal{S})\, e_k = \begin{bmatrix} -e_1^T C^{(k-1)} e_{k-1} \\ \vdots \\ -e_{k-1}^T C^{(k-1)} e_{k-1} \\ 1 \\ 0 \\ \vdots \end{bmatrix}, \qquad k = 2, \ldots, n, \tag{2.2}$$

then the unique upper Hessenberg matrix $H(\mathcal{R})$ with the entry one along the subdiagonal and with the spectrum $\lambda_1, \ldots, \lambda_n$ such that the kth leading principal submatrix has eigenvalues $\rho_1^{(k)}, \ldots, \rho_k^{(k)}$ for all $k = 1, \ldots, n-1$ is

$$H(\mathcal{R}) = U(\mathcal{S})^{-1} C^{(n)} U(\mathcal{S}). \tag{2.3}$$

Proof. We will show that the spectrum of the $k \times k$ leading principal submatrix of $H(\mathcal{R})$ is $\rho_1^{(k)}, \ldots, \rho_k^{(k)}$ (uniqueness of $H(\mathcal{R})$ was shown in [36] and will also be proved later). Let U_k denote the $k \times k$ leading principal submatrix of $U(\mathcal{S})$, and let, for $j > k$, $\tilde{u}_j$ denote the vector of the first k entries of the jth column of $U(\mathcal{S})^{-1}$. The spectrum of the $k \times k$ leading principal submatrix of $H(\mathcal{R})$ is the spectrum of

$$[I_k, 0]\, U(\mathcal{S})^{-1} C^{(n)} U(\mathcal{S}) \begin{bmatrix} I_k \\ 0 \end{bmatrix} = [U_k^{-1}, \tilde{u}_{k+1}, \ldots, \tilde{u}_n] \begin{bmatrix} 0 \\ U_k \\ 0 \end{bmatrix} = [U_k^{-1}, \tilde{u}_{k+1}] \begin{bmatrix} 0 \\ U_k \end{bmatrix}.$$

It is also the spectrum of the matrix

$$U_k [U_k^{-1}, \tilde{u}_{k+1}] \begin{bmatrix} 0 \\ U_k \end{bmatrix} U_k^{-1} = [I_k, U_k \tilde{u}_{k+1}] \begin{bmatrix} 0 \\ I_k \end{bmatrix},$$

which is a companion matrix with last column $U_k \tilde{u}_{k+1}$. From

$$\begin{aligned} e_{k+1} = U_{k+1} U_{k+1}^{-1} e_{k+1} &= \begin{bmatrix} U_k & -C^{(k)} e_k \\ 0 & 1 \end{bmatrix} \begin{bmatrix} U_k^{-1} & \tilde{u}_{k+1} \\ 0 & 1 \end{bmatrix} e_{k+1} \\ &= \begin{bmatrix} U_k \tilde{u}_{k+1} - C^{(k)} e_k \\ 1 \end{bmatrix} \end{aligned}$$

we obtain $U_k \tilde{u}_{k+1} = C^{(k)} e_k$. ∎

Note that (2.3) represents a similarity transformation separating the spectrum of $H(\mathcal{R})$ from the strict Ritz values $\mathcal{S}$ of $H(\mathcal{R})$. The matrix $U(\mathcal{S})$ transforms the companion matrix whose strict Ritz values are all zero to a Hessenberg matrix with arbitrary Ritz values, and it is itself composed of (parts of) companion matrices. We will call $U(\mathcal{S})$, for lack of a better name, the *Ritz value companion transform.*

Clearly, the Ritz values generated in the Arnoldi method can exhibit any convergence behavior: It suffices to apply the Arnoldi process with the initial Arnoldi vector e_1 and the matrix $H(\mathcal{R})$ with arbitrary prescribed Ritz values from Proposition 2.1. Then the method generates the Hessenberg matrix $H(\mathcal{R})$ itself. If the prescribed Ritz values occur in complex conjugate pairs, then the Ritz value companion transform $U(\mathcal{S})$ and the Hessenberg matrix $H(\mathcal{R})$ in (2.3) are real, and the Arnoldi process runs without complex arithmetics.

We next look for a parametrization of the class of all matrices and initial Arnoldi vectors generating given Ritz values. From $H(\mathcal{R})$ we can easily obtain an upper Hessenberg matrix whose leading principal submatrices have the same prescribed eigenvalues but with arbitrary positive values along the subdiagonal. Let $\sigma_1, \sigma_2, \ldots, \sigma_{n-1}$ be given positive real numbers, and consider the similarity transformation

$$H \equiv \operatorname{diag}(1, \sigma_1, \sigma_1\sigma_2, \ldots, \Pi_{j=1}^{n-1}\sigma_j)\, H(\mathcal{R}) \left(\operatorname{diag}(1, \sigma_1, \sigma_1\sigma_2, \ldots, \Pi_{j=1}^{n-1}\sigma_j)\right)^{-1}.$$

Then the subdiagonal of H has the entries $\sigma_1, \sigma_2, \ldots, \sigma_{n-1}$, and all leading principal submatrices of H are similar to the corresponding leading principal submatrices of $H(\mathcal{R})$. The following theorem shows the uniqueness of H.

THEOREM 2.2. *Let the set*

$$\begin{aligned}\mathcal{R} = \{\, & \rho_1^{(1)}, \\ & (\rho_1^{(2)}, \rho_2^{(2)}), \\ & \vdots \\ & (\rho_1^{(n-1)}, \ldots, \rho_{n-1}^{(n-1)}), \\ & (\lambda_1, \ldots\ldots\ldots, \lambda_n)\,\}\end{aligned}$$

represent any choice of $n(n+1)/2$ *complex Ritz values, and let*

$$D_\sigma = \operatorname{diag}(1, \sigma_1, \sigma_1\sigma_2, \ldots, \Pi_{j=1}^{n-1}\sigma_j),$$

where $\sigma_1, \sigma_2, \ldots, \sigma_{n-1}$ *are* $n-1$ *positive real numbers. Then*

$$H = D_\sigma\, H(\mathcal{R}) D_\sigma^{-1}$$

is the unique Hessenberg matrix H *with subdiagonal entries*

$$h_{k+1,k} = \sigma_k, \qquad k = 1, \ldots, n-1,$$

with eigenvalues $\lambda_1, \ldots, \lambda_n$ *and with* $\rho_1^{(k)}, \ldots, \rho_k^{(k)}$ *being the eigenvalues of its* k*th leading principal submatrix for all* $k = 1, \ldots, n-1$.

Proof. We have already explained that H has the desired Ritz values and subdiagonal entries. It remains to show uniqueness. For this we need a recursion for the characteristic polynomials of the leading submatrices H_k of H. We denote the prescribed characteristic polynomial of H_k by $p_k(\lambda)$, and by $\sigma^{k,i}$ we denote the product of prescribed subdiagonal entries

$$\sigma^{k,i} = \prod_{\ell=i}^{k} \sigma_\ell.$$

We also define the polynomial $p_0(\lambda) \equiv 1$. Using expansion along the last column to compute the determinant of $H_k - \lambda I$, we get

$$\begin{aligned}\det(H_k - \lambda I) = {} & (-1)^{k-1} h_{1,k} \sigma^{k-1,1} + (-1)^{k-2} h_{2,k} p_1(\lambda) \sigma^{k-1,2} \\ & + (-1)^{k-3} h_{3,k} p_2(\lambda) \sigma^{k-1,3} + \cdots + (h_{k,k} - \lambda) p_{k-1}(\lambda),\end{aligned}$$

and hence we have the recursion

$$p_k(\lambda) = (h_{kk} - \lambda) p_{k-1}(\lambda) + \sum_{i=1}^{k-1} (-1)^{k-i} h_{ik} \sigma^{k-1,i} p_{i-1}(\lambda), \quad 1 \le k \le n. \tag{2.4}$$

Now assume that both H and $\tilde{H}$ have the desired Ritz values and subdiagonal entries, and let us prove that $H = \tilde{H}$ by induction for all subsequent leading principal submatrices. Clearly, $h_{1,1} = \tilde{h}_{1,1} = \rho_1^{(1)}$, and if the claim is valid for all leading principal submatrices of dimension at most $k-1$, then the entries of H_k and $\tilde{H}_k$ can differ only in the last column. By comparing the coefficients (subsequently before λ^k until λ^0) of the polynomial $p_k(\lambda)$ as given in (2.4) with the coefficients given by

$$p_k(\lambda) = (\tilde{h}_{kk} - \lambda) p_{k-1}(\lambda) + \sum_{i=1}^{k-1} (-1)^{k-i} \tilde{h}_{ik} \sigma^{k-1,i} p_{i-1}(\lambda),$$

we obtain $h_{ik} = \tilde{h}_{ik}$ subsequently for $i = k, k-1, \ldots, 1$. ∎

Theorem 2.2 immediately leads to a parametrization of the matrices and initial Arnoldi vectors that generate a given set of Ritz values $\mathcal{R}$. In addition, the subdiagonal of the generated Hessenberg matrix can be prescribed.

COROLLARY 2.3. *Assume that we are given a set of tuples of complex numbers*

$$\begin{aligned}\mathcal{R} = \{&\rho_1^{(1)},\\ &(\rho_1^{(2)}, \rho_2^{(2)}),\\ &\vdots\\ &(\rho_1^{(n-1)}, \ldots, \rho_{n-1}^{(n-1)}),\\ &(\lambda_1, \ldots\ldots\ldots, \lambda_n)\}\end{aligned}$$

and $n-1$ positive real numbers $\sigma_1, \ldots, \sigma_{n-1}$. If A is a matrix of order n and b a nonzero n-dimensional vector, then the following assertions are equivalent:

1. *The Hessenberg matrix generated by the Arnoldi process applied to A and initial Arnoldi vector b has eigenvalues $\lambda_1, \ldots, \lambda_n$ and subdiagonal entries $\sigma_1, \ldots, \sigma_{n-1}$, and $\rho_1^{(k)}, \ldots, \rho_k^{(k)}$ are the eigenvalues of its kth leading principal submatrix for all $k = 1, \ldots, n-1$.*
2. *The matrix A is of the form*

$$A = V D_\sigma U(\mathcal{S})^{-1} C^{(n)} U(\mathcal{S}) D_\sigma^{-1} V^*$$

and $b = \|b\| V e_1$, where V is a unitary matrix, D_σ is the diagonal matrix

$$D_\sigma = \operatorname{diag}(1, \sigma_1, \sigma_1\sigma_2, \ldots, \Pi_{j=1}^{n-1}\sigma_j),$$

$U(\mathcal{S})$ is the Ritz value companion transform in (2.2), and $C^{(n)}$ is the companion matrix of the polynomial with roots $\lambda_1, \ldots, \lambda_n$.

Corollary 2.3 is an analogue of Theorem 1.1 on arbitrary convergence of the GMRES method. Here we prescribe k values (the k Ritz values) in the kth iteration, whereas Theorem 1.1 prescribes one value (the kth residual norm); the spectrum of A is prescribed in both results. Note that in [43] it was shown that if the Arnoldi method produces a *particular* sequence of $n(n+1)/2$ Ritz values, the same sequence can be generated by a whole class of matrices together with initial Arnoldi vectors. The paper also gives a description of this class. It can be seen as an analogue of the earlier result of Greenbaum and Strakoš [22], showing that if a residual norm convergence curve is generated by GMRES, the same curve can be obtained by a whole class of matrices together with right-hand sides. Our corollary shows, surprisingly, that for general nonnormal matrices the distribution of the Ritz values generated in the Arnoldi method can be arbitrary and fully independent of the spectrum. We remark that there exist some results on the distribution of Ritz values for specific nonnormal matrices, for example, for Jordan blocks and block diagonal matrices with a simple normal eigenvalue; see [7].

The given parametrization may give some additional insight into the convergence behavior of versions of Arnoldi used in practice, e.g., implicitly restarted Arnoldi with polynomial shifts [4, 5]; in particular it may help one to better understand cases where Arnoldi with exact shifts fails; see, e.g., [13]. As Ritz values are contained in the field of values, it may also have implications for field of values–based analysis of iterative methods.

We deal here with the problem of constructing both an input matrix and an initial vector to produce prescribed Ritz values. In Corollary 2.3 the initial vector

$b = \|b\|Ve_1$ could be chosen arbitrarily if we define A appropriately, since the only requirement for the matrix V is to be unitary. When the matrix A is given, changing b will, of course, change the Ritz values. Constructing an initial vector to produce prescribed Ritz values was done for the Lanczos method in [41]. If a Hermitian matrix has distinct eigenvalues, that paper shows how to construct a perverse initial vector such that the Ritz values in the next-to-last iteration are as far from the eigenvalues as allowed by the interlacing property (see [14] for a generalization to the normal case).

Another consequence of Corollary 2.3 is that the Ritz values in the Arnoldi method are in general independent of the subdiagonal elements $h_{k+1,k}$ of the generated Hessenberg matrix. This is not that strange if one realizes that $h_{k+1,k}$ is not an element of the matrix H_k used to extract the current Ritz values. But, on the other hand, the independency from $h_{k+1,k}$ is still surprising in view of the fact that one is used to regarding the residual norm

$$\|AV^{(k)}y - \rho^{(k)}V^{(k)}y\| = h_{k+1,k}|e_k^T y| \tag{2.5}$$

for an eigenpair $(\rho^{(k)}, y)$ of H_k (see (2.1)) as a measure of the quality of the approximate Ritz value-vector pair $(\rho^{(k)}, V^{(k)}y)$. Corollary 2.3 shows that, in theory, any small nonzero value of $h_{k+1,k}$ is possible with $\rho^{(k)}$ arbitrarily far from the eigenvalues of A. And conversely, all eigenvalues of H_k may coincide with eigenvalues of A with an arbitrarily large value of $h_{k+1,k}$. Though it is known that the residual norm is not always indicative for the quality of the Ritz values (see, e.g., [8, 18]), one might expect that in such counterintuitive cases, the misleading behavior of $h_{k+1,k}$ is compensated for by $|e_k^T y|$ in (2.5). But consider the following: Let A be parametrized as $A = VH(\mathcal{R})V^*$ and $b = Ve_1$, and let for an approximate Ritz value-vector pair $(\rho^{(k)}, V^{(k)}y)$ the residual norm in (2.5) be $|e_k^T y|$ (all subdiagonal entries $h_{k+1,k}$ of $H(\mathcal{R})$ are one), where

$$H(\mathcal{R})_k y = \rho^{(k)} y.$$

For any choice of small nonzero entries $\sigma_1, \ldots, \sigma_{n-1}$, the matrix $VD_\sigma H(\mathcal{R})D_\sigma^{-1}V^*$ with $D_\sigma = \mathrm{diag}(1, \sigma_1, \ldots, \Pi_{j=1}^{n-1}\sigma_j)$ generates the same Ritz value $\rho^{(k)}$, but the residual norm in (2.5) will change as $\sigma_k|e_k^T y_s|$, where

$$\left(D_{\sigma_k} H(\mathcal{R})_k D_{\sigma_k}^{-1}\right) y_s = \rho^{(k)} y_s$$

with $D_{\sigma_k} = \mathrm{diag}(1, \sigma_1, \ldots, \Pi_{j=1}^{k-1}\sigma_j)$. However, the eigenvector y_s is nothing but a scaling of y because

$$\left(D_{\sigma_k} H(\mathcal{R})_k D_{\sigma_k}^{-1}\right) (D_{\sigma_k} y) = \rho^{(k)} (D_{\sigma_k} y),$$

i.e., $y_s = D_{\sigma_k} y$. This means that, with appropriate subdiagonal entries, the value $|e_k^T y_s|$ can be small *too* (even if y_s is normalized) and does not compensate for a small σ_k, in spite of a possibly diverging Ritz value $\rho^{(k)}$. Something similar can be said about cases where all eigenvalues of H_k coincide with eigenvalues of A for arbitrarily large values of σ_k.

3. Prescribed convergence behavior of the Arnoldi and the GMRES methods for the same pair $\{A, b\}$. The diagonal matrix D_σ with positive entries in Corollary 2.3 contains the subdiagonal entries of the generated Hessenberg matrix, and it can be chosen arbitrarily for any prescribed Ritz values. Because the values of these subdiagonal entries influence the residual norms generated by the GMRES

method applied to the corresponding linear system, there is a chance we can modify the behavior of GMRES while maintaining the prescribed Ritz values. This is what we will investigate next. Rather than directly choosing the diagonal matrix D_σ to control GMRES convergence, we will derive an alternative parametrization of the matrices and initial Arnoldi vectors that generate a given set of Ritz values. This parametrization will reveal the relation with the parametrization in Theorem 1.1 and thus might enable us to combine prescribing Ritz values with prescribing GMRES residual norms.

The parametrization in Corollary 2.3 is based on a unitary matrix V whose columns span the nth Krylov subspace $\mathcal{K}_n(A,b)$, whereas the parametrization in Theorem 1.1 works with a unitary matrix W whose columns span $A\mathcal{K}_n(A,b)$. To better understand the relation between Corollary 2.3 and Theorem 1.1, we will translate the former parametrization in terms of the latter. To achieve this, we will use two factorizations of the Krylov matrix

$$K \equiv \left[b, Ab, A^2b, \ldots, A^{n-1}b\right],$$

one with V and one with W. The first factorization is nothing but the QR decomposition

$$K = VU \tag{3.1}$$

of K. By *the* QR decomposition we will always mean the unique QR decomposition whose upper triangular factor has positive real main diagonal. The upper triangular factor U is related to the generated Ritz values as follows.

LEMMA 3.1. *Let H be the Hessenberg matrix generated by an Arnoldi process terminating at the nth iteration applied to A and b, and let $U(\mathcal{S})$ be the Ritz value companion transform in* (2.2) *corresponding to the generated strict Ritz values. Then the upper triangular factor U of the QR factorization* (3.1) *of the Krylov matrix K is*

$$U = \|b\| \operatorname{diag}\left(1, h_{2,1}, h_{2,1}h_{3,2}, \ldots, \Pi_{j=1}^{n-1} h_{j+1,j}\right) U(\mathcal{S})^{-1}.$$

Proof. Any Arnoldi process (terminating at the nth iteration) can be written according to the parametrization of Corollary 2.3 with $D_\sigma = \operatorname{diag}(1, h_{2,1}, \ldots, \Pi_{j=1}^{n-1} h_{j+1,j})$. Then in the Krylov matrix

$$K = \left[b, Ab, \ldots, A^{n-1}b\right]$$

we can take $\|b\|V$ out of the brackets to factor it since

$$\begin{aligned}
b &= \|b\| V e_1, \\
Ab &= \|b\| V D_\sigma U(\mathcal{S})^{-1} C^{(n)} U(\mathcal{S}) D_\sigma^{-1} e_1, \\
A^2b &= \|b\| V \left(D_\sigma U(\mathcal{S})^{-1} C^{(n)} U(\mathcal{S}) D_\sigma^{-1}\right)^2 e_1, \\
\cdots &= \cdots \\
A^{n-1}b &= \|b\| V \left(D_\sigma U(\mathcal{S})^{-1} C^{(n)} U(\mathcal{S}) D_\sigma^{-1}\right)^{n-1} e_1.
\end{aligned}$$

Therefore

$$K = \|b\| V \left[e_1, D_\sigma U(\mathcal{S})^{-1} C^{(n)} U(\mathcal{S}) D_\sigma^{-1} e_1, \ldots, \left(D_\sigma U(\mathcal{S})^{-1} C^{(n)} U(\mathcal{S}) D_\sigma^{-1}\right)^{n-1} e_1\right].$$

Now we would like to show that the last matrix on the right-hand side is just $D_\sigma U(\mathcal{S})^{-1}$. The first entry of the diagonal matrix D_σ being one, we have $U(\mathcal{S})D_\sigma^{-1}e_1 = e_1$. Obviously we have $(D_\sigma U(\mathcal{S})^{-1}C^{(n)}U(\mathcal{S})D_\sigma^{-1})^j = (D_\sigma U(\mathcal{S})^{-1}(C^{(n)})^j U(\mathcal{S})D_\sigma^{-1})$. Hence $\left(D_\sigma U(\mathcal{S})^{-1}C^{(n)}U(\mathcal{S})D_\sigma^{-1}\right)^j e_1 = D_\sigma U(\mathcal{S})^{-1}(C^{(n)})^j e_1$. It is straightforward to see that $(C^{(n)})^j e_1 = e_{j+1}$. This yields

$$\left(D_\sigma U(\mathcal{S})^{-1}C^{(n)}U(\mathcal{S})D_\sigma^{-1}\right)^j e_1 = D_\sigma U(\mathcal{S})^{-1}e_{j+1}, \quad j = 0, \ldots, n-1,$$

and hence we have the factorization $K = \|b\|VD_\sigma U(\mathcal{S})^{-1}$. On the other hand, $K = VU$. The uniqueness of the QR factorization gives $U = \|b\|D_\sigma U(\mathcal{S})^{-1}$. ∎

A similar result is proved in [28, Proposition 3.1]. The second factorization of K which we need involves the unitary factor W. We prove the following result in the same way as the previous lemma; it was also proved in [2] in a different way.

LEMMA 3.2. *Consider a matrix A with initial Arnoldi vector b such that the Arnoldi process does not terminate before iteration n. If $A = WYC^{(n)}Y^{-1}W^*$ and $b = Wh$ according to Theorem* 1.1*, then we have*

$$K = WY.$$

Proof. With Theorem 1.1 the Krylov matrix is defined as

$$K = \left[Wh, AWh, A^2Wh, \ldots, A^{n-1}Wh\right].$$

We wish to take W out of the brackets to factor K. This can be done since

$$\begin{aligned} AW &= WYC^{(n)}Y^{-1}, \\ A^2W &= W(YC^{(n)}Y^{-1})^2, \\ \cdots &= \cdots \\ A^{n-1}W &= W(YC^{(n)}Y^{-1})^{n-1}. \end{aligned}$$

Therefore

$$K = W\left[h, YC^{(n)}Y^{-1}h, \ldots, (YC^{(n)}Y^{-1})^{n-1}h\right].$$

Now we would like to show that the last matrix on the right-hand side is just Y. The vector h being the first column of Y, we have $h = Ye_1$. Obviously we have $(YC^{(n)}Y^{-1})^j = Y(C^{(n)})^jY^{-1}$. Hence $(YC^{(n)}Y^{-1})^j h = Y(C^{(n)})^j e_1$. As before, $(C^{(n)})^j e_1 = e_{j+1}$. This yields

$$(YC^{(n)}Y^{-1})^j h = Ye_{j+1}, \quad j = 0, \ldots, n-1,$$

and this proves the result. ∎

With the two factorizations $K = VU = WY$ we are ready for a second parametrization, formulated with the notation of Theorem 1.1 and based on the unitary matrix W, of the pairs $\{A, b\}$ generating arbitrary Ritz values.

THEOREM 3.3. *Assume that we are given a set of tuples of complex numbers*

$$\begin{aligned} \mathcal{R} = \{&\rho_1^{(1)}, \\ &(\rho_1^{(2)}, \rho_2^{(2)}), \\ &\vdots \\ &(\rho_1^{(n-1)}, \ldots, \rho_{n-1}^{(n-1)}), \\ &(\lambda_1, \ldots\ldots\ldots, \lambda_n)\}, \end{aligned}$$

such that $(\lambda_1, \ldots, \lambda_n)$ *contains only nonzero numbers, and* $n-1$ *positive real numbers* $\sigma_1, \ldots, \sigma_{n-1}$. *If* A *is a matrix of order* n *and* b *a nonzero* n*-dimensional vector, then the following assertions are equivalent:*

1. *The Hessenberg matrix generated by the Arnoldi process applied to* A *and initial Arnoldi vector* b *has eigenvalues* $\lambda_1, \ldots, \lambda_n$ *and subdiagonal entries* $\sigma_1, \ldots, \sigma_{n-1}$, *and* $\rho_1^{(k)}, \ldots, \rho_k^{(k)}$ *are the eigenvalues of its* k*th leading principal submatrix for all* $k = 1, \ldots, n-1$.
2. *The matrix* A *is of the form*

$$A = WYC^{(n)}Y^{-1}W^*$$

and $b = Wh$, *where* W *is a unitary matrix,* $C^{(n)}$ *is the companion matrix corresponding to the eigenvalues* $\lambda_1, \ldots, \lambda_n$, *and* Y *is of the form*

$$Y = \begin{bmatrix} h & R \\ 0 & \end{bmatrix}.$$

R is the upper triangular matrix

$$R = \Gamma L^* T \tag{3.2}$$

of order $n-1$, *where* T *is the trailing principal submatrix in the partitioning*

$$\|b\| \operatorname{diag}(1, \sigma_1, \sigma_1\sigma_2, \ldots, \Pi_{j=1}^{n-1}\sigma_j) U(\mathcal{S})^{-1} = \begin{bmatrix} \|b\| & t^* \\ 0 & T \end{bmatrix} \tag{3.3}$$

of the scaled inverse of the Ritz value companion transform $U(\mathcal{S})$ *in* (2.2) *and* L *is the lower triangular factor in the Cholesky decomposition*

$$LL^* = I_{n-1} + T^{-*}tt^*T^{-1}. \tag{3.4}$$

The diagonal matrix Γ *with unit modulus entries is such that*

$$e_k^T \Gamma L^{-1} T^{-*} t \geq 0, \quad k = 1, \ldots, n-1, \tag{3.5}$$

and the entries of $h = [\eta_1, \ldots, \eta_n]^T$ *satisfy*

$$[\eta_1, \ldots, \eta_{n-1}]^T = \|b\| \, \Gamma L^{-1} T^{-*} t, \qquad \eta_n = \|b\| \sqrt{1 - \|L^{-1}T^{-*}t\|^2}. \tag{3.6}$$

Proof. First we prove the implication $1 \to 2$. Because the Arnoldi process does not stop before the last iteration, GMRES applied to the linear system with matrix A, right-hand side b, and zero initial guess does not stop before the last iteration, and we can write $A = WYC^{(n)}Y^{-1}W^*$ and $b = Wh$ according to Theorem 1.1. From Lemma 3.2, the factorization (3.1), and Lemma 3.1, we have

$$K^*K = Y^*W^*WY = Y^*Y, \qquad K^*K = U^*V^*VU = \|b\|^2 U(\mathcal{S})^{-*} D_\sigma^T D_\sigma U(\mathcal{S})^{-1}.$$

Hence the matrix Y from the parametrization must satisfy

$$Y^*Y = \|b\|^2 U(\mathcal{S})^{-*} D_\sigma^T D_\sigma U(\mathcal{S})^{-1}.$$

Let $\hat{h} = [\eta_1, \ldots, \eta_{n-1}]^T$ be the vector of the first $n-1$ components of h from (1.2). Then from (1.1) we have

$$Y^*Y = \begin{bmatrix} \|h\|^2 & \hat{h}^*R \\ R^*\hat{h} & R^*R \end{bmatrix}. \tag{3.7}$$

Comparing (3.7) with $\|b\|^2 U(\mathcal{S})^{-*} D_\sigma^T D_\sigma U(\mathcal{S})^{-1}$ and using the partitioning (3.3), we obtain for R and $\hat{h}$ the conditions

$$R^*R = T^*T + tt^*, \qquad \hat{h} = \|b\| R^{-*} t. \tag{3.8}$$

Furthermore, we have the conditions $\eta_k \geq 0$, $k = 1, \ldots, n-1$, because all entries of $\hat{h}$ correspond to entries describing the GMRES convergence curve according to (1.2).

Let L be the lower triangular factor in the Cholesky decomposition

$$LL^* = I_{n-1} + T^{-*} t t^* T^{-1},$$

let Γ be a diagonal matrix with unit modulus entries, and let $R = \Gamma L^* T$. Then

$$R^*R = T^* L \Gamma^* \Gamma L^* T = T^*(I_{n-1} + T^{-*} t t^* T^{-1}) T = T^*T + tt^*$$

is always satisfied and Γ can be chosen such that

$$e_k^T \Gamma L^{-1} T^{-*} t \geq 0, \quad k = 1, \ldots, n-1.$$

It follows that

$$\hat{h} = \|b\| R^{-*} t = \|b\| \, \Gamma L^{-1} T^{-*} t,$$

and with $\|h\| = \|W^* b\| = \|b\|$ we obtain

$$\eta_n = \sqrt{\|h\|^2 - \|\hat{h}\|^2} = \|b\| \sqrt{1 - \|L^{-1} T^{-*} t\|^2}.$$

For the implication $2 \to 1$, let $A = WYC^{(n)} Y^{-1} W^*$ be the parametrization of A given in assertion 2, and let $b = Wh$. By Lemma 3.2, $K = WY$; let $K = V\tilde{U}$ be the QR factorization of the Krylov matrix K. We first show that $\tilde{U} = \|b\| D_\sigma U(\mathcal{S})^{-1}$.

In the QR decomposition $K = V\tilde{U}$ we have $Ve_1 = b/\|b\|$, and therefore we can partition $\tilde{U}$ as

$$\tilde{U} = \begin{bmatrix} \|b\| & \tilde{t}^* \\ 0 & \tilde{T} \end{bmatrix}. \tag{3.9}$$

With the first part of the proof

$$R^*R = \tilde{T}^* \tilde{T} + \tilde{t}\tilde{t}^*, \qquad \hat{h} = \|b\| R^{-*} \tilde{t},$$

(see (3.8)), i.e.,

$$\tilde{t} = \frac{R^* \hat{h}}{\|b\|}, \qquad \tilde{T}^* \tilde{T} = R^*R - \frac{R^* \hat{h} \hat{h}^* R}{\|b\|^2}.$$

But by assumption, we have for t and T from (3.4) and (3.6) the same equalities,

$$t = \frac{T^* L \Gamma^* \hat{h}}{\|b\|} = \frac{R^* \hat{h}}{\|b\|},$$

$$T^*T = T^*(LL^* - T^{-*} t t^* T^{-1}) T = T^* L \Gamma^* \Gamma L^* T - tt^* = R^*R - \frac{R^* \hat{h} \hat{h}^* R}{\|b\|^2}.$$

The matrix $R^*R - \frac{R^*\hat{h}\hat{h}^*R}{\|b\|^2}$ is positive definite since it is the Schur complement of $\|h\|^2$ in Y^*Y, which is positive definite. Therefore the Cholesky decomposition of the matrix $R^*R - \frac{R^*\hat{h}\hat{h}^*R}{\|b\|^2}$ exists, and $\tilde{T} = T$ is the unique Cholesky factor. Together with $\tilde{t} = t = \frac{R^*\hat{h}}{\|b\|}$ we have

$$\tilde{U} = \|b\| D_\sigma U(\mathcal{S})^{-1}.$$

Because of $K = WY = V\tilde{U}$ and with (2.3) it follows that

$$\begin{aligned} A &= WYC^{(n)}Y^{-1}W^* = V\tilde{U}C^{(n)}\tilde{U}^{-1}V^* \\ &= VD_\sigma U(\mathcal{S})^{-1}C^{(n)}U(\mathcal{S})D_\sigma^{-1}V^* = VD_\sigma H(\mathcal{R})D_\sigma^{-1}V^*. \end{aligned}$$

The upper Hessenberg matrix $D_\sigma H(\mathcal{R})D_\sigma^{-1}$ generated by the Arnoldi method therefore has the prescribed Ritz values and subdiagonal entries. ∎

Note that Theorem 3.3 and Corollary 2.3 are not fully equivalent. In Theorem 3.3 we must assume, for reasons of compatibility with Theorem 1.1, that the spectrum of A does not contain the origin. In Corollary 2.3 the only free parameters are a unitary matrix and the norm of the initial Arnoldi vector. In Theorem 3.3 there appears to be slightly more freedom because a unit modulus entry of Γ can lie anywhere on the unit circle if the corresponding entry of $L^{-1}T^{-*}t$ is zero; see (3.5). There is of course much less freedom in Theorem 3.3 than there is in the parametrization of Theorem 1.1 when prescribing a GMRES convergence curve.

We see that by modifying the choice of the subdiagonal entries $\sigma_1, \ldots, \sigma_{n-1}$ in Theorem 3.3, we might modify the vector h representing the GMRES convergence curve generated with A and b while maintaining the prescribed Ritz values, i.e., while leaving the Ritz value companion transform $U(\mathcal{S})$ in (3.3) unchanged. Does this mean we can force any GMRES convergence speed with arbitrary Ritz values? There is one situation where this is certainly not possible: When there is a zero Ritz value in some iteration, this implies a singular Hessenberg matrix and corresponds to an indefinable iterate in the full orthogonalization method, which is equivalent to stagnation in the parallel GMRES process; see, e.g., [6, 19]. Hence zero Ritz values are equivalent with GMRES stagnation. For completeness, we give another proof of this well-known fact, formulated with the notation of Theorem 3.3.

LEMMA 3.4. *With the notation of Theorem* 3.3 *and for* $1 \le k \le n-1$, *the* k*-tuple* $(\rho_1^{(k)}, \ldots, \rho_k^{(k)})$ *contains a zero Ritz value if and only if* $\eta_k = 0$ *in* (3.6).

Proof. Denote by $U(\mathcal{S})$ the Ritz value companion transform in (2.2), and let it be partitioned according to (3.3) as

$$U(\mathcal{S}) = \|b\| D_\sigma \begin{bmatrix} \|b\| & t^* \\ 0 & T \end{bmatrix}^{-1} = \|b\| D_\sigma \begin{bmatrix} \frac{1}{\|b\|} & \frac{-t^*T^{-1}}{\|b\|} \\ 0 & T^{-1} \end{bmatrix},$$

where $D_\sigma = \operatorname{diag}(1, \sigma_1, \sigma_1\sigma_2, \ldots, \Pi_{j=1}^{n-1}\sigma_j)$. By definition of $U(\mathcal{S})$, the k-tuple $(\rho_1^{(k)}, \ldots, \rho_k^{(k)})$ contains a zero Ritz value if and only if $t^*T^{-1}e_k = 0$. It can easily be checked that the lower triangular factor L in the Cholesky decomposition

$$LL^* = I_{n-1} + T^{-*}tt^*T^{-1}$$

has its kth row and column zero, except for the diagonal entry, if and only if $t^*T^{-1}e_k = 0$. Then the vector $\hat{h}$, being the solution of the lower triangular system

$$L\Gamma^*\hat{h} = T^{-*}t,$$

has kth entry zero if and only if $t^*T^{-1}e_k = 0$. ∎

Thus GMRES residual norms cannot be fully independent of Ritz values. However, we will show that the *only* restriction Ritz values put on GMRES residual norms is precisely that zero Ritz values imply stagnation. Otherwise, any GMRES behavior is possible with arbitrary prescribed Ritz values. Before proving this, we need the following auxiliary result.

LEMMA 3.5. *Consider n positive real numbers*

$$f(0) \geq f(1) \geq \cdots \geq f(n-1) > 0,$$

and define

$$\eta_k = (f(k-1)^2 - f(k)^2)^{1/2}, \quad k < n, \quad \eta_n = f(n-1), \qquad \hat{h} = [\eta_1, \ldots, \eta_{n_1}]^T.$$

If we denote by R_h the upper triangular factor of the Cholesky decomposition

$$R_h^T R_h = I_{n-1} - \frac{\hat{h}\hat{h}^T}{f(0)^2},$$

then we have

$$e_k^T R_h^{-T} \hat{h} = 0 \quad \Leftrightarrow \quad f(k-1) = f(k), \qquad k = 1, \ldots, n-1.$$

Proof. The entries of R_h^T are

(3.10)

$$(R_h^T)_{i,k} = \frac{-\eta_i \eta_k}{\sqrt{\eta_{k+1}^2 + \cdots + \eta_n^2}\sqrt{\eta_k^2 + \cdots + \eta_n^2}}, \quad (R_h^T)_{k,k} = \frac{\sqrt{\eta_{k+1}^2 + \cdots + \eta_n^2}}{\sqrt{\eta_k^2 + \cdots + \eta_n^2}};$$

see [17] on the Cholesky decomposition of a rank-one updated identity matrix, or also [29, Theorem 4.2]. Therefore, if $\eta_k = 0$ for some $k \leq n-1$, then the kth row and kth column of R_h^T are zero except for the main diagonal entry. It is easily seen from solving the lower triangular system $R_h^T x = \hat{h}$ with forward substitution that $x = R_h^{-T}\hat{h}$ is zero. ∎

THEOREM 3.6. *Consider a set of tuples of complex numbers*

$$\begin{aligned} \mathcal{R} = \{ & \rho_1^{(1)}, \\ & (\rho_1^{(2)}, \rho_2^{(2)}), \\ & \vdots \\ & (\rho_1^{(n-1)}, \ldots, \rho_{n-1}^{(n-1)}), \\ & (\lambda_1, \ldots\ldots\ldots, \lambda_n)\}, \end{aligned}$$

such that $(\lambda_1, \ldots, \lambda_n)$ contains no zero number, and n positive numbers

$$f(0) \geq f(1) \geq \cdots \geq f(n-1) > 0,$$

such that $f(k-1) = f(k)$ if and only if the k-tuple $(\rho_1^{(k)}, \ldots, \rho_k^{(k)})$ contains a zero number. Let A be a square matrix of size n, and let b be a nonzero n-dimensional vector. The following assertions are equivalent:

1. *The GMRES method applied to A and right-hand side b with zero initial guess yields residuals $r^{(k)}$, $k = 0, \ldots, n-1$, such that*

$$\|r^{(k)}\| = f(k), \quad k = 0, \ldots, n-1,$$

A has eigenvalues $\lambda_1, \ldots, \lambda_n$, and $\rho_1^{(k)}, \ldots, \rho_k^{(k)}$ are the eigenvalues of the kth leading principal submatrix of the generated Hessenberg matrix for all $k = 1, \ldots, n-1$.

2. *The matrix A is of the form*

$$A = WYC^{(n)}Y^{-1}W^*$$

and $b = Wh$, where W is a unitary matrix and $C^{(n)}$ is the companion matrix corresponding to the polynomial with roots $\lambda_1, \ldots, \lambda_n$. Y is given by

$$Y = \begin{bmatrix} h & R \\ & 0 \end{bmatrix},$$

h being the vector

$$h = [\eta_1, \ldots, \eta_n]^T, \qquad \eta_k = (f(k-1)^2 - f(k)^2)^{1/2}, \quad k < n, \quad \eta_n = f(n-1),$$

and R being the nonsingular upper triangular matrix of order $n-1$

$$R = R_h^{-1} D_c^{-*} C^{-1}, \tag{3.11}$$

where C is the trailing principal submatrix in the partitioning

$$U(\mathcal{S}) = \begin{bmatrix} 1 & c^* \\ 0 & C \end{bmatrix} \tag{3.12}$$

of the Ritz value companion transform $U(\mathcal{S})$ for $\mathcal{R}$ defined in (2.2)*. R_h is the upper triangular factor of the Cholesky decomposition*

$$R_h^T R_h = I_{n-1} - \frac{\hat{h}\hat{h}^T}{f(0)^2}$$

for $\hat{h} = [\eta_1, \ldots, \eta_{n-1}]^T$, and D_c is a nonsingular diagonal matrix such that

$$R_h^{-T}\hat{h} = -f(0)^2 D_c\, c. \tag{3.13}$$

Proof. Because of Theorem 1.1 it is clear that the parametrization given here generates the prescribed GMRES residual norms and vice versa. Hence it suffices to show that the given parametrization generates the prescribed Ritz values and vice versa. For this we will use the parametrization of Theorem 3.3 and prove that the matrix R in (3.11) satisfies the same conditions as the upper triangular R in (3.2) in Theorem 3.3.

First we show that the nonsingular diagonal matrix D_c used to define R in (3.11) exists. With the assumed partitioning (3.12) of $U(\mathcal{S})$ and by the definition of $U(\mathcal{S})$, the entries of c are zero precisely at positions corresponding to iterations with a zero Ritz value. By assumption, $\hat{h}$ is zero at exactly these positions and so is $R_h^{-T}\hat{h}$ with Lemma 3.5. Thus we can always define a nonsingular diagonal matrix D_c such that

$$R_h^{-T}\hat{h} = -f(0)^2 D_c c.$$

Now with the definition (3.11) of R we have

$$R^*\hat{h} = -f(0)^2 C^{-*} c.$$

Next, in analogy with (3.3), consider the partitioning

$$\operatorname{diag}(f(0), D_c^{-*})\, U(\mathcal{S})^{-1} = \begin{bmatrix} f(0) & t^* \\ 0 & T \end{bmatrix} \tag{3.14}$$

of a diagonal scaling of $U(\mathcal{S})^{-1} = \left[\begin{smallmatrix} 1 & -c^*C^{-1} \\ 0 & C^{-1} \end{smallmatrix}\right]$. It follows that

$$t = -f(0)C^{-*}c = \frac{R^*\hat{h}}{f(0)}$$

and

$$T = D_c^{-*} C^{-1}.$$

To prove that the matrix R in (3.11) satisfies the same conditions as the upper triangular R in (3.2) in Theorem 3.3, it remains to show that $R_h^{-1} = L^*$, $\Gamma = I_{n-1}$, where L and Γ are the matrices defined in the second assertion of Theorem 3.3. We have

$$\begin{aligned} I_{n-1} + T^{-*} t t^* T^{-1} &= I_{n-1} + D_c C^* \frac{R^*\hat{h}}{f(0)} \left(D_c C^* \frac{R^*\hat{h}}{f(0)} \right)^* \\ &= I_{n-1} + \frac{R_h^{-T}\hat{h}}{f(0)} \left(\frac{R_h^{-T}\hat{h}}{f(0)} \right)^* = R_h^{-T} \left(R_h^T R_h + \frac{\hat{h}\hat{h}^*}{f(0)^2} \right) R_h^{-1} \\ &= R_h^{-T} R_h^{-1} \end{aligned}$$

and with $\Gamma = I_{n-1}$

$$e_k^T R_h^T T^{-*} t = e_k^T R_h^T \frac{R_h^{-T}\hat{h}}{f(0)} = \frac{\eta_k}{f(0)} \geq 0, \quad k = 1, \ldots, n-1.$$

Together with

$$\begin{aligned} \eta_n = f(n-1) &= \sqrt{f(0)^2 - (f(0)^2 - f(1)^2) - \cdots - (f(n-2)^2 - f(n-1)^2)} \\ &= f(0)\sqrt{1 - \frac{\|\hat{h}\|^2}{f(0)^2}}, \end{aligned}$$

we have that matrices of the form

$$W \begin{bmatrix} h & R \\ 0 & \end{bmatrix} C(\mathcal{R}^{(n)}) \begin{bmatrix} h & R \\ 0 & \end{bmatrix}^{-1} W^*$$

and right-hand sides Wh generate the prescribed Ritz values and vice versa; see Theorem 3.3. ∎

The only freedom we have to prescribe both Ritz values and GMRES residual norms is in the unitary matrix W and in those entries of the diagonal matrix D_c on positions corresponding to iterations with a zero Ritz value or, equivalently, on positions corresponding to iterations where GMRES stagnates. On these positions D_c may have arbitrary values. In this sense we have exhausted all the degrees of

freedom; GMRES and Arnoldi are invariant under unitary transformation, and more values than Ritz values and residual norms cannot be prescribed for the same Arnoldi process.

Theorem 3.6 says that one can construct matrices and right-hand sides for which converged Ritz values need not imply accelerated convergence speed in the GMRES method, as is the case for the CG method for Hermitian positive definite matrices [45]. The only restriction Ritz values put on GMRES is that a zero Ritz value leads to stagnation in the corresponding iteration. A restricted role of Ritz values for GMRES may be expected in view of the fact that the Ritz values are *not* the roots of the polynomials GMRES generates to compute its residuals. These roots are the harmonic Ritz values [35, 19]. Although harmonic Ritz values generated in the Arnoldi procedure might be prescribed in a way similar to what we did for ordinary Ritz values in the previous section [30], it is not clear whether this is possible with given GMRES residual norms. Nonetheless, the extent to which ordinary Ritz values and residual norms are independent is astonishing. Note, for example, that for matrices close to normal the bounds derived in [46] suggest that as soon as eigenvalues of such matrices are sufficiently well approximated by Ritz values, GMRES from then on converges at least as fast as for a related system in which these eigenvalues are missing. This may be surprising, but it is not contradictory.

Note that we also could have formulated the second assertion in the previous theorem analogously to the second assertion in Theorem 3.3. Then the diagonal scaling matrix in (3.3) takes the form of the diagonal matrix in (3.14); otherwise the assertion need not be changed. Translated in the notation of Corollary 2.3, this gives the following alternative parametrization.

COROLLARY 3.7. *Assume that we are given a set of tuples of complex numbers*

$$\begin{aligned}\mathcal{R} = \{&\rho_1^{(1)},\\ &(\rho_1^{(2)}, \rho_2^{(2)}),\\ &\vdots\\ &(\rho_1^{(n-1)}, \ldots, \rho_{n-1}^{(n-1)}),\\ &(\lambda_1, \ldots\ldots\ldots, \lambda_n)\},\end{aligned}$$

such that $(\lambda_1, \ldots, \lambda_n)$ *contains no zero number, and* n *positive real numbers*

$$f(0) \geq f(1) \geq \cdots \geq f(n-1) > 0,$$

such that $f(k-1) = f(k)$ *if and only if the* k*-tuple* $(\rho_1^{(k)}, \ldots, \rho_k^{(k)})$ *contains a zero number. If* A *is a matrix of order* n *and* b *a nonzero* n*-dimensional vector, then the following assertions are equivalent:*

1. *The GMRES method applied to* A *and right-hand side* b *with zero initial guess yields residuals* $r^{(k)}$, $k = 0, \ldots, n-1$, *such that*

$$\|r^{(k)}\| = f(k), \quad k = 0, \ldots, n-1,$$

   A has eigenvalues $\lambda_1, \ldots, \lambda_n$, *and* $\rho_1^{(k)}, \ldots, \rho_k^{(k)}$ *are the eigenvalues of the* k*th leading principal submatrix of the generated Hessenberg matrix for all* $k = 1, \ldots, n-1$.
2. *The matrix A is of the form*

$$A = V \operatorname{diag}(f(0), D_c^{-*})\, U(\mathcal{S})^{-1} C^{(n)} U(\mathcal{S}) \operatorname{diag}(f(0)^{-1}, D_c^*) V^*$$

and $b = \|b\|Ve_1$, where V is a unitary matrix, $U(\mathcal{S})$ is the Ritz value companion transform for $\mathcal{R}$ defined in (2.2)*, and $C^{(n)}$ is the companion matrix of the polynomial with roots $\lambda_1, \ldots, \lambda_n$. D_c is a nonsingular diagonal matrix such that*

$$R_h^{-T}\hat{h} = -f(0)^2 D_c c$$

with $\hat{h}$ being the vector

$$\hat{h} = [\eta_1, \ldots, \eta_{n-1}]^T, \qquad \eta_k = (f(k-1)^2 - f(k)^2)^{1/2},$$

R_h being the upper triangular factor of the Cholesky decomposition

$$R_h^T R_h = I_{n-1} - \frac{\hat{h}\hat{h}^T}{f(0)^2},$$

and c being the first row of $U(\mathcal{S})$ without its diagonal entry.

This parametrization is based on unitary matrices V spanning $\mathcal{K}_n(A, b)$ instead of unitary matrices W spanning $A\mathcal{K}_n(A, b)$ and is therefore closer to the actual Arnoldi process which is run in standard implementations of the GMRES and Arnoldi methods. On the other hand, the parametrization in Theorem 3.6 reveals more clearly the relation with the prescribed residual norms. Note that we can easily change Corollary 3.7 to yield a "V-based" analogue of Theorem 1.1; it suffices to consider $U(\mathcal{S})$ as a free parameter matrix. Corollary 3.7 also shows how to define the subdiagonal entries $h_{k+1,k}$ of a Hessenberg matrix with prescribed Ritz values in order to obtain prescribed GMRES residual norms: They follow from the equality

$$f(0)\,\mathrm{diag}\left(1, h_{2,1}, h_{2,1}h_{3,2}, \ldots, \Pi_{j=1}^{n-1} h_{j+1,j}\right) = \mathrm{diag}\,(f(0), D_c^{-*}).$$

4. Conclusions and future work. The Arnoldi orthogonalization process is a cornerstone of several successful Krylov subspace methods for non-Hermitian matrices. Nevertheless, two of the most popular methods based on it, the GMRES and the Arnoldi methods, can exhibit counterintuitive convergence behavior. For GMRES it has been known for some time that any nonincreasing convergence curve is possible and can be generated with any spectrum [21]; the fact that *all* Ritz values formed by the Arnoldi method can be prescribed appears not to have been noticed so far. The present paper also shows that arbitrary convergence of GMRES is possible not only with any spectrum, but even with any Ritz values for all iterations (provided that we treat the stagnation case correctly).

Given the success of (modified versions of) the GMRES and Arnoldi methods for a large variety of problems, the situations described in our theoretical results may occur rarely in solving practical problems in scientific computing. For example, in the Arnoldi method, cases of Ritz values diverging further away from the spectrum in every iteration are possible, as we proved in section 2, but they happen for particular matrices only in combination with particular initial Arnoldi vectors. As one normally chooses the initial Arnoldi vector randomly, the chances that this vector produces diverging Ritz values may be small, and in practice one can easily rerun the process with a different random initial Arnoldi vector. In the GMRES method, however, one is stuck with a given right-hand side, and applications exist where the pathological cases described in [21] occur. An example is given by convection-diffusion problems; see, e.g., [37] or [26, Figures 3.10 and 3.11]. This type of problem also contains an illustration of our results of section 3: In the convection dominated case, system matrices are

often close to transposed Jordan blocks (i.e., upper Hessenberg matrices with identical Ritz values for all iterations), and, for certain boundary conditions, right-hand sides are close to the first unit vector [26]. Hence we have almost converged Ritz values from the very start, but this does not mean that GMRES converges rapidly as one would expect. On the contrary, it is known that these problems give very slow, nearly stagnating GMRES residual norms during the initial phase of convergence [15, 26].

It is often assumed that counterintuitive GMRES behavior, i.e., spectral information which is misleading for residual norms, is possible in the highly nonnormal case only, and one may expect the counterintuitive results of this paper to be restricted to the highly nonnormal case, too. Neither of the two statements is entirely correct; for instance, arbitrary GMRES convergence curves are possible for such nice normal matrices as are the perfectly conditioned unitary matrices; see [22, section 3.1] and [21]. As for our results on the Arnoldi method, certainly prescribed Ritz values outside the convex hull of the eigenvalues are possible with nonnormal matrices only, and probably the further one prescribes Ritz values away from the convex hull, the more nonnormal the constructed input matrix must be. On the other hand, divergence *inside* the convex hull might still be possible with some normal but non-Hermitian matrices. Very little appears to the authors to be known on this topic (for general normal matrices of size three, see, e.g., [7]). Although there are generalized interlacing properties for normal matrices, they cannot be exploited because the leading principal submatrices of normal Hessenberg matrices need not be normal. Let us also recall that the Ritz values generated in the Lanczos method in the next-to-last iteration can be as far from the eigenvalues as allowed by the interlacing property [41].

Our results are of a theoretical nature and may give additional insight into the properties of the GMRES and the Arnoldi methods. An important issue related to our results is how to detect, a priori, whether a matrix with initial vector will lead to diverging Ritz value behavior in Arnoldi or to stagnation in GMRES. For GMRES, work on complete or partial stagnation was done, for example, in [48] or, recently, in [29], where the results are linked with the parametrization in Theorem 1.1. More generally, the question is whether our theory gives some insight into what is a good Arnoldi starting vector, respectively, right-hand side b. Work for the near future includes modifications of our results for popular restarted versions of Arnoldi or GMRES which may enhance theoretical insight into the behavior of strategies that are frequently used in practice.

Software. At http://www.cs.cas.cz/duintjertebbens/duintjertebbens_soft.html the reader can find MATLAB subroutines to create matrices and initial vectors with the parametrizations in this paper.

Acknowledgments. The authors are indebted to Zdeněk Strakoš for initiating their work on this topic. They thank the anonymous referees for their comments, and they thank Russel Carden for pointing out reference [43].

REFERENCES

[1] G. S. Ammar and C. Y. He, *On an inverse eigenvalue problem for unitary Hessenberg matrices*, Linear Algebra Appl., 218 (1995), pp. 263–271.

[2] M. Arioli, V. Pták, and Z. Strakoš, *Krylov sequences of maximal length and convergence of GMRES*, BIT, 38 (1998), pp. 636–643.

[3] W. E. Arnoldi, *The principle of minimized iteration in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.

[4] C. Beattie, M. Embree, and J. Rossi, *Convergence of restarted Krylov subspaces to invariant subspaces*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 1074–1109.
[5] C. A. Beattie, M. Embree, and D. C. Sorensen, *Convergence of polynomial restart Krylov methods for eigenvalue computations*, SIAM Rev., 47 (2005), pp. 492–515.
[6] P. N. Brown, *A theoretical comparison of the Arnoldi and GMRES algorithms*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 58–78.
[7] R. Carden, *Ritz Values and Arnoldi Convergence for Non-Hermitian Matrices*, Ph.D. thesis, Rice University, Houston, TX, 2011.
[8] F. Chatelin, *Eigenvalues of Matrices*, John Wiley & Sons, Chichester, UK, 1993.
[9] G. N. de Oliveira, *Matrices with prescribed characteristic polynomial and a prescribed submatrix*. I, Pacific J. Math. 29 (1969), pp. 653–661.
[10] G. N. de Oliveira, *Matrices with prescribed characteristic polynomial and a prescribed submatrix*. II, Pacific J. Math. 29 (1969), pp. 663–667.
[11] M. Eiermann, *Fields of values and iterative methods*, Linear Algebra Appl., 180 (1993), pp. 167–197.
[12] M. Eiermann and O. G. Ernst, *Geometric aspects of the theory of Krylov subspace methods*, Acta Numer., 10 (2001), pp. 251–312.
[13] M. Embree, *The Arnoldi eigenvalue iteration with exact shifts can fail*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 1–10.
[14] T. Ericsson, *On the Eigenvalues and Eigenvectors of Hessenberg Matrices*, Numerical Analysis Group, Göteborg, Report 10, Chalmers University of Technology and the University of Göteborg, Department of Computer Sciences, Göteborg, Sweden, 1990; available online at http://www.math.chalmers.se/Math/Research/NumericalAnalysis/num_analysis/reports/ericsson_Hessenberg_matrices.ps.gz.
[15] O. G. Ernst, *Residual-minimizing Krylov subspace methods for stabilized discretizations of convection-diffusion equations*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1079–1101.
[16] K. Fan and G. Pall, *Imbedding conditions for Hermitian and normal matrices*, Canad. J. Math., 9 (1957), pp. 298–304.
[17] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders, *Methods for modifying matrix factorizations*, Math. Comp., 28 (1974), pp. 505–535.
[18] S. Godet-Thobie, *Eigenvalues of Large Highly Nonnormal Matrices*, Ph.D. thesis, University Paris IX, Dauphine, Paris, France, 1993.
[19] S. Goossens and D. Roose, *Ritz and harmonic Ritz values and the convergence of FOM and GMRES*, Numer. Linear Algebra Appl., 6 (1999), pp. 281–293.
[20] A. Greenbaum, *Generalizations of the field of values useful in the study of polynomial functions of a matrix*, Linear Algebra Appl., 347 (2002), pp. 233–249.
[21] A. Greenbaum, V. Pták, and Z. Strakoš, *Any nonincreasing convergence curve is possible for GMRES*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 465–469.
[22] A. Greenbaum and Z. Strakoš, *Matrices that generate the same Krylov residual spaces*, in Recent Advances in Iterative Methods, IMA Vol. Math. Appl. 60, Springer, New York, 1994, pp. 95–118.
[23] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp. 409–436.
[24] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Standards, 45 (1950), pp. 255–282.
[25] C. Lanczos, *Solution of systems of linear equations by minimized iterations*, J. Res. Nat. Bur. Standards, 49 (1952), pp. 33–53.
[26] J. Liesen and Z. Strakoš, *GMRES convergence analysis for a convection-diffusion model problem*, SIAM J. Sci. Comput., 26 (2005), pp. 1989–2009.
[27] S. M. Malamud, *Inverse spectral problem for normal matrices and the Gauss-Lucas theorem*, Trans. Amer. Math. Soc., 357 (2005), pp. 4043–4064.
[28] G. Meurant, *GMRES and the Arioli, Pták and Strakoš parametrization*, BIT, 52 (2012), pp. 687–702.
[29] G. Meurant, *Necessary and sufficient conditions for GMRES complete and partial stagnation*, submitted.
[30] G. Meurant, *Notes on GMRES Convergence* (15): *The Matrix H and the Ritz Values*, private communication, 2011.
[31] G. Meurant and Z. Strakoš, *The Lanczos and conjugate gradient algorithms in finite precision arithmetic*, Acta Numer., 15 (2006), pp. 471–542.
[32] C. C. Paige, *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*, Ph.D. thesis, University of London, London, UK, 1971.
[33] C. C. Paige, *Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix*, J. Inst. Math. Appl., 18 (1976), pp. 341–349.

[34] C. C. Paige, *Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem*, Linear Algebra Appl., 34 (1980), pp. 235–258.
[35] C. C. Paige, B. N. Parlett, and H. A. van der Vorst, *Approximate solutions and eigenvalue bounds from Krylov subspaces*, Numer. Linear Algebra Appl., 2 (1995), pp. 115–133.
[36] B. N. Parlett and G. Strang, *Matrices with prescribed Ritz values*, Linear Algebra Appl., 428 (2008), pp. 1725–1739.
[37] S. C. Reddy and L. N. Trefethen, *Pseudospectra of the convection-diffusion operator*, SIAM J. Appl. Math., 54 (1994), pp. 1634–1649.
[38] Y. Saad, *Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices*, Linear Algebra Appl., 34 (1980), pp. 269–295.
[39] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Algorithms Archit. Adv. Sci. Comput., Manchester University Press, Manchester, UK, 1992.
[40] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[41] D. S. Scott, *How to make the Lanczos algorithm converge slowly*, Math. Comp., 33 (1979), pp. 239–247.
[42] N. Shomron and B. N. Parlett, *Linear algebra meets Lie algebra: The Kostant-Wallach theory*, Linear Algebra Appl., 431 (2009), pp. 1745–1767.
[43] P. Smit, *Generating Identical Ritz Values*, Research Memorandum FEW 696, Faculty of Economics and Business Administration, Tilburg University, Tilburg, The Netherlands, 1995.
[44] L. N. Trefethen and M. Embree, *Spectra and Pseudospectra*, Princeton University Press, Princeton, NJ, 2005.
[45] A. van der Sluis and H. A. van der Vorst, *The rate of convergence of conjugate gradients*, Numer. Math., 48 (1986), pp. 543–560.
[46] H. A. van der Vorst and C. Vuik, *The superlinear convergence behaviour of GMRES*, J. Comput. Appl. Math., 48 (1993), pp. 327–341.
[47] E. Vecharynski and J. Langou, *Any admissible cycle-convergence behavior is possible for restarted GMRES at its initial cycles*, Numer. Linear Algebra Appl., 18 (2011), pp. 499–511.
[48] I. Zavorin, D. P. O'Leary, and H. Elman, *Complete stagnation of GMRES*, Linear Algebra Appl., 367 (2003), pp. 165–183.
[49] J.-P. M. Zemke, *Hessenberg eigenvalue-eigenmatrix relations*, Linear Algebra Appl., 414 (2006), pp. 589–606.

Linear Algebra and its Applications 450 (2014) 83–107

Contents lists available at ScienceDirect

Linear Algebra and its Applications

www.elsevier.com/locate/laa

On investigating GMRES convergence using unitary matrices

J. Duintjer Tebbens [a,b,*], G. Meurant [c], H. Sadok [d], Z. Strakoš [e]

[a] *Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 2, 182 07 Praha 8, Libeň, Czech Republic*
[b] *Faculty of Pharmacy in Hradec Králové, Charles University in Prague, Heyrovského 1203, 500 05 Hradec Králové, Czech Republic*
[c] *30 rue du sergent Bauchat, 75012 Paris, France*
[d] *Laboratoire de Mathématiques Pures et Appliquées, Université du Littoral, 62228 Calais Cedex, France*
[e] *Faculty of Mathematics and Physics, Charles University in Prague, Sokolovská 83, 186 75 Prague, Czech Republic*

ARTICLE INFO

Article history:
Received 21 March 2013
Accepted 26 February 2014
Available online xxxx
Submitted by D.B. Szyld

MSC:
65F10

Keywords:
GMRES convergence
Unitary matrices
Unitary spectra
Normal matrices
Krylov residual subspace
Schur parameters

ABSTRACT

For a given matrix A and right-hand side b, this paper investigates unitary matrices generating, with some right-hand sides c, the same GMRES residual norms as the pair (A, b). We give characterizations of this class of unitary matrices and point out the relationship with Krylov subspaces and Krylov residual subspaces for the pair (A, b). We investigate the eigenvalues of these unitary matrices in relation to the convergence behavior of GMRES for the pair (A, b) and describe the indispensable role of the eigenvector information. We conclude with a formula for the GMRES residual norms generated by a normal matrix B in terms of its eigenvalues and components of the right-hand side c in the eigenvector basis.

* Corresponding author at: Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 2, 182 07 Praha 8, Libeň, Czech Republic. Tel.: +420 266 052 182.
E-mail addresses: duintjertebbens@cs.cas.cz (J. Duintjer Tebbens), gerard.meurant@gmail.com (G. Meurant), hassane.sadok@lmpa.univ-littoral.fr (H. Sadok), strakos@karlin.mff.cuni.cz (Z. Strakoš).

http://dx.doi.org/10.1016/j.laa.2014.02.044
0024-3795/

1. Introduction

In this paper we consider the convergence behavior of the GMRES method for solving linear systems

$$Ax = b$$

with (generally complex) square matrices A of order n and right-hand sides b; for a detailed description of this popular Krylov subspace method see [1] or [2]. With no loss of generality, we consider zero initial guess $x_0 = 0$. The kth GMRES iterate is the vector x_k in the kth Krylov subspace which minimizes the residual norm, that is

$$x_k = \arg \min_{x \in \mathcal{K}_k(A,b)} \|b - Ax\|, \quad \mathcal{K}_k(A,b) \equiv \operatorname{span}\{b, Ab, \ldots, A^{k-1}b\}. \tag{1}$$

It follows that the kth residual vector $r_k = b - Ax_k$ is the difference between b and its orthogonal projection onto the Krylov *residual* subspace $A\mathcal{K}_k(A,b)$. A standard convergence bound for the kth residual norm with diagonalizable A is

$$\frac{\|r_k\|}{\|b\|} \leqslant \kappa(Z) \min_{p \in \Pi_k} \max_{i=1,\ldots,n} |p_k(\lambda_i)|, \tag{2}$$

where A has the spectral decomposition $A = Z\Lambda Z^{-1}$, $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, $\kappa(Z)$ is the condition number of the eigenvector matrix and Π_k is the set of polynomials of degree k with the value one at the origin (see, e.g., [2]). For Hermitian matrices, convergence of Krylov subspace methods like Conjugate Gradient or MINRES is very strongly linked with the eigenvalue distribution. For instance, the values these methods minimize (some norm of the error or residual vector) can be bounded with the same bound as in (2) where $\kappa(Z) = 1$. This bound then depends on the spectrum only and, concerning the envelope of all possible convergence curves for matrices A having the given spectrum, it is sharp [3], i.e., for every k there exists a right-hand side (depending on k) such that equality holds in (2). However, it has been known for some time that eigenvalues alone cannot explain GMRES convergence for non-Hermitian and, more specifically, for non-normal matrices. This was first shown in the 1994 paper [4], in which the authors studied the matrices B that generate the same Krylov residual space as the one given by the pair (A,b), that is

$$B\mathcal{K}_k(B,b) = A\mathcal{K}_k(A,b), \quad k = 1, 2, \ldots, n.$$

Then GMRES applied to (B,b) yields the same convergence history (with respect to residual norms) as GMRES applied to (A,b). Matrices B with this property will be called GMRES(A,b)-equivalent matrices. They can be characterized as follows (we assume for simplicity of notation, that GMRES applied to (A,b) does not terminate until the step n, i.e., $\dim(\mathcal{K}_n(A,b)) = n$).

Theorem 1. *(See Theorem 1 of [4].) Let W be a unitary matrix whose first k columns give a basis of $A\mathcal{K}_k(A,b)$ for all k with $1 \leqslant k \leqslant n$ and let $\mathcal{H}$ be an unreduced upper Hessenberg matrix such that $AW = W\mathcal{H}$. Then, the following assertions are equivalent:*

1. *B is GMRES(A,b)-equivalent.*
2. *$B = W\tilde{R}\mathcal{H}W^*$, where $\tilde{R}$ is any nonsingular upper triangular matrix.*

Among other results, it is shown in [4] that the spectrum of B can consist of arbitrary nonzero values. In [5] this was extended by proving the fact that any nonincreasing sequence of residual norms can be generated by GMRES and [6] closed this series of papers with a full parametrization of the class of matrices and right-hand sides giving prescribed convergence history while the system matrix has prescribed nonzero spectrum; for a survey we refer to [7, Section 5.7]. In [8] a parametrization was given of the class of matrices and right-hand sides generating, in addition to prescribed residual norms and eigenvalues, prescribed Ritz values in all iterations.

While all these results show that spectral information only can be misleading when explaining GMRES convergence behavior with general matrices, GMRES convergence *is* bounded using the eigenvalue distribution when it is applied to normal matrices in view of (2). More strongly, GMRES convergence is for normal matrices *determined* by the approximation problem

$$\|r_k\| = \min_{p\in\Pi_k} \|p(\Lambda)Z^*b\|. \tag{3}$$

On p. 105 of [4] the authors wrote, with respect to Theorem 1: "If, for each vector b, we can find a matrix B of the given form, for which we can analyze the behavior of the GMRES method applied to B, then we can also analyze the behavior of the GMRES method applied to A. Since the behavior of the GMRES method for normal matrices is well-understood in terms of the eigenvalues of the matrix, it is desirable to find an upper triangular matrix $\tilde{R}$ such that $\tilde{R}\mathcal{H}$ is normal." It was shown subsequently in [4] that $\tilde{R}$ can always be chosen such that $\tilde{R}\mathcal{H}$ is normal and even unitary, and under some assumptions such that $\tilde{R}\mathcal{H}$ is Hermitian positive definite or just Hermitian. In general, however, no simple properties of A were found which are related to the spectral properties of a GMRES(A,b)-equivalent normal matrix.

Two papers, both published in 2000, analyzed the eigenvalues of particular unitary GMRES(A,b)-equivalent matrices and studied in detail the relation with GMRES convergence. In [9], Liesen used QR and RQ factorizations of the matrix $\mathcal{H}$ to obtain bounds for the residual norms in terms of the largest gap in the spectrum of the Q factors on the unit circle. He showed, among other things, that a large maximum gap in the spectrum of the Q factor in an RQ factorization $\mathcal{H} = R\mathcal{Q}$ implies fast GMRES convergence; see also his PhD thesis [10, Section 5]. In [11], Knizhnerman considered $\mathcal{H}$ to be a possibly infinite dimensional bounded operator and showed an inverse result,

namely that for finite operators (matrices), fast GMRES convergence implies a large gap in the spectrum of $\mathcal{Q}$ in a certain RQ factorization of $\mathcal{H}$. It was also shown that the entries of this particular $\mathcal{Q}$ can be expressed in terms of the residual norms only [11, Section 6.1].

The goal of this paper is to further explore how and to what extent GMRES convergence can be explained using unitary GMRES(A, b)-equivalent pairs. With a unitary GMRES(A, b)-equivalent pair (B, c) we mean a matrix B with a right-hand side c which generate the same convergence history as (A, b) where B is unitary and c is not necessarily equal to b. We will characterize such pairs and investigate their properties. Unlike in [4], we will not consider the unitary matrix W in $B = W\tilde{R}\mathcal{H}W^*$ (see Theorem 1) merely an unimportant change of variables matrix, whose influence is not taken into account when analyzing the spectrum of B. Our investigation will rely on the fact that the first k columns of W form a basis of $A\mathcal{K}_k(A, b)$, $1 \leqslant k \leqslant n$. We show that all unitary GMRES(A, b)-equivalent matrices B can be constructed from W and from a unitary matrix V whose first k orthogonal columns form bases of $\mathcal{K}_k(A, b)$, $1 \leqslant k \leqslant n$. Since V and W depend strongly on the interplay between A and b, our goal cannot be in relating GMRES convergence to some simple properties of A only. Instead, we will describe how both the eigenvalues of B and components of c in the eigenvector basis of B determine the convergence curve. This will be based on a new explicit expression for the kth residual norm generated by a normal matrix.

The paper is organized as follows. Section 2 characterizes GMRES(A, b)-equivalent matrices B and pairs (B, c), where B is unitary, and it explains their relationship to the Krylov subspaces and Krylov residual subspaces produced by A and b. In this section it is also shown that the eigenvectors of B play a substantial role in the description of convergence for GMRES applied to (A, b). Section 3 contains the derivation of a formula for the kth GMRES residual norm in terms of the eigenvalues, eigenvectors and the right-hand side when GMRES is applied to a normal matrix. These quantities can be useful to gain some insight in the convergence for (A, b) with the help of a GMRES(A, b)-equivalent pair where the matrix is normal and, in particular, unitary.

Throughout the paper we will assume, as mentioned above, that GMRES does not terminate until the last step n. Hence, the Krylov subspaces are of full dimension and their orthogonal bases constructed using the Gram–Schmidt algorithm are well defined. We also assume the zero initial guess in all applications of GMRES. For simplicity we normalize the right-hand side b such that $\|b\| = 1$. We will use repeatedly the fact that GMRES residual norm convergence is unitarily invariant in the sense that, U being any unitary matrix, the pair (U^*AU, U^*b) generates the same residual norms as (A, b). With e_i we will denote the ith column of the identity matrix (of appropriate order). With "the subdiagonal" and "subdiagonal entries" of an upper Hessenberg matrix we will mean the (entries on the) first subdiagonal under the main diagonal. Hessenberg matrices with a real positive subdiagonal will be denoted with a plus as lower index, for example H_+.

2. Unitary GMRES(A, b)-equivalent pairs

In this section we characterize pairs (B, c) which yield the same GMRES convergence history as (A, b) with B unitary and c not necessarily equal to b. We describe their relationship to the Krylov subspaces and Krylov residual subspaces generated by (A, b) and study the influence of the spectrum of B on the convergence history for (A, b).

2.1. Unitary GMRES(A, b)-equivalent matrices

First let us consider the case where $c = b$. Here is a characterization of the class of all unitary GMRES(A, b)-equivalent matrices.

Proposition 2. *Let W be a unitary matrix whose first k columns give a basis of $A\mathcal{K}_k(A, b)$ for $1 \leqslant k \leqslant n$, let $\mathcal{H}$ be an unreduced upper Hessenberg matrix such that $AW = W\mathcal{H}$ and let $\mathcal{H} = R\mathcal{Q}$ be an RQ factorization of $\mathcal{H}$. Then, the following assertions are equivalent:*

1. *B is unitary and GMRES(A, b)-equivalent.*
2. *$B = WD_1\mathcal{Q}W^*$, where D_1 is a diagonal unitary matrix.*

Proof. Any RQ factorization of $\mathcal{H}$ has the form $\mathcal{H} = (RD_0^{-1})(D_0\mathcal{Q})$, where D_0 is a diagonal unitary matrix. Thus, with the notation of Theorem 1, B is unitary if and only if $B = W(\tilde{R}RD_0^{-1})(D_0\mathcal{Q})W^*$ is unitary, which is true if and only if $\tilde{R}R$ is unitary. This holds if and only if $\tilde{R} = D_1R^{-1}$ for a diagonal unitary matrix D_1, giving $B = WD_1D_0^{-1}D_0\mathcal{Q}W^* = WD_1\mathcal{Q}W^*$. □

Thus *all* unitary GMRES(A, b)-equivalent matrices are of the form $B = W\mathcal{Q}W^*$ where $\mathcal{Q}$ is the unitary factor of an RQ decomposition of $\mathcal{H}$.

Note that the Hessenberg matrix $\mathcal{H}$ is not the Hessenberg matrix generated in a standard implementation of GMRES where an orthogonal basis of $\mathcal{K}_n(A, b)$ is built. $\mathcal{H}$ results from building an orthogonal basis of $A\mathcal{K}_n(A, b)$ by starting the Arnoldi process with the vector $Ab/\|Ab\|$. This is done, for example, in the Walker–Zhou implementation of GMRES [12]. In a standard implementation of GMRES, one constructs the unitary matrix $\hat{V}$ whose first k columns span the Krylov space $\mathcal{K}_k(A, b)$ for all $1 \leqslant k \leqslant n$ and which is the result of the Arnoldi orthogonalization process applied to (A, b). More precisely, the unitary $\hat{V}$ satisfies

$$A\hat{V} = \hat{V}H_+, \qquad \hat{V}e_1 = b, \tag{4}$$

for an unreduced upper Hessenberg matrix H_+ with positive subdiagonal entries. Consider the unique QR decomposition

$$H_+ = Q^+ \mathcal{R} \tag{5}$$

such that Q^+ is a unitary upper Hessenberg matrix with a real positive first row; see [13]. The entries of the matrix $\mathcal{Q}$ from an RQ decomposition of $\mathcal{H}$ were given in terms of the GMRES residual norms in Eq. (6.1) of [11]. Interestingly enough, the moduli of the entries of Q^+ and $\mathcal{Q}$ coincide. In order to show this, we need the following lemma. For real matrices, it was also proved in [13, Theorem 3.1].

Lemma 3. *Let W be a unitary matrix whose first k columns give a basis of $A\mathcal{K}_k(A,b)$ for $1 \leqslant k \leqslant n$ and let $\hat{V}$ be the unitary matrix in (4). If Q^+ is the unitary factor in the QR factorization (5) of H_+, then*

$$Q^+ = \hat{V}^* W D_2,$$

where D_2 is a diagonal unitary matrix.

Proof. Because the first k columns of W form a basis of $A\mathcal{K}_k(A,b)$, $1 \leqslant k \leqslant n$, we can write

$$A\hat{V} = W\hat{R}$$

for some nonsingular upper triangular matrix $\hat{R}$. Then from $A\hat{V} = \hat{V}H_+ = W\hat{R}$ we have the QR decomposition

$$H_+ = \big(\hat{V}^* W\big)\hat{R}.$$

Hence, for the properly chosen diagonal unitary matrix D_2, $Q^+ = \hat{V}^* W D_2$ has its first row real and positive. □

Corollary 4. *Let Q^+ be the unitary factor in the QR factorization (5) and let $\mathcal{Q}$ be the unitary factor of an RQ decomposition of $\mathcal{H}$. Then*

$$\mathcal{Q} = D_3^* Q^+ D_2^*$$

where D_2 is the matrix of Lemma 3 and D_3 is a diagonal unitary matrix.

Proof. From (4) and Lemma 3 we have

$$AWD_2\big(Q^+\big)^* = WD_2\big(Q^+\big)^* H_+,$$

which implies

$$W^*AW = \mathcal{H} = D_2\big(Q^+\big)^* H_+ Q^+ D_2^*.$$

Hence we have the RQ decomposition

$$\mathcal{H} = D_2\big(Q^+\big)^*\big(Q^+\mathcal{R}\big)Q^+D_2^* = (D_2\mathcal{R})\big(Q^+D_2^*\big).$$

Therefore, $\mathcal{Q}$ is of the form $\mathcal{Q} = D_3^*Q^+D_2^*$ for some diagonal unitary matrix D_3. □

Lemma 3 enables another characterization of unitary GMRES(A, b)-equivalent matrices; cf. Proposition 2.

Theorem 5. *The following assertions are equivalent:*

1. *B is unitary and GMRES(A, b)-equivalent.*
2. *$B = WV^*$, where V is a unitary matrix whose first k columns give a basis of $\mathcal{K}_k(A, b)$ and W is a unitary matrix whose first k columns give a basis of $A\mathcal{K}_n(A, b)$ for $1 \leqslant k \leqslant n$.*

Proof. Because of Proposition 2, if B is unitary and GMRES(A, b)-equivalent, then B is of the form

$$B = \hat{W}D_1\mathcal{Q}\hat{W}^*,$$

where the columns of $\hat{W}$ are an orthonormal basis of $A\mathcal{K}_n(A, b)$. Using Lemma 3 and Corollary 4, we obtain

$$B = \hat{W}D_1\mathcal{Q}\hat{W}^* = \hat{W}D_1D_3^*Q^+D_2^*\hat{W}^* = \hat{W}D_1D_3^*\hat{V}^*\hat{W}\hat{W}^* = \hat{W}D_1D_3^*\hat{V}^*.$$

Putting $V = \hat{V}D_3$ and $W = \hat{W}D_1$ gives the first implication. Now let $B = WV^*$. Then with Lemma 3, for some diagonal unitary matrix D_4,

$$B = W\big(V^*W\big)W^* = W\big(D_4\hat{V}^*W\big)W^* = W\big(D_4Q^+D_2^*\big)W^*$$

and with Corollary 4,

$$B = W\big(D_4Q^+D_2^*\big)W^* = W\big(D_4D_3\mathcal{Q}D_2D_2^*\big)W^* = W(D_4D_3\mathcal{Q})W^*.$$

This yields the second implication if we use Proposition 2. □

This theorem shows how closely unitary GMRES(A, b)-equivalent matrices are related to the Krylov subspaces $\mathcal{K}_k(A, b)$ and the Krylov residual subspaces $A\mathcal{K}_k(A, b)$ for $1 \leqslant k \leqslant n$. These subspaces, and therefore also the matrices V and W, depend strongly on the interplay between A and b. Linking the properties of unitary GMRES(A, b)-equivalent matrices (like spectral properties) to some simple properties of A only is therefore rather complicated.

With the help of Theorem 5 we can characterize the eigenvalues of unitary GMRES(A, b)-equivalent matrices in terms of the Krylov subspaces $\mathcal{K}_k(A, b)$ and the Krylov residual subspaces $A\mathcal{K}_k(A, b)$, $1 \leqslant k \leqslant n$. GMRES convergence for (A, b) is bounded by these eigenvalues in the following sense.

Corollary 6. *Using the notation of Theorem 5, the GMRES residual norms for the pair (A, b) are bounded as*

$$\frac{\|r_k\|}{\|b\|} \leqslant \min_{p\in\Pi_k} \max_{i=1,\ldots,n} |p_k(\mu_i)|, \tag{6}$$

with $\mu_1, \ldots, \mu_n$ being the eigenvalues in the generalized eigenvalue problem

$$V^*x = \mu W^*x. \tag{7}$$

It is worth noticing that in (6) the polynomials are evaluated at points which depend through V and W in (7) on the right-hand side b.

2.2. Unitary GMRES(A, b)-equivalent pairs

Now we come to unitary matrices that give the same residual norm convergence curve as (A, b) with a right-hand side possibly different from b. Our goal will be to characterize the set of all pairs (B, c) with these properties. *Some* pairs are obtained simply by using the fact that GMRES convergence is unitarily invariant. For example, let us consider a unitary GMRES(A, b)-equivalent matrix $B = WV^*$ defined in Theorem 5 and let us define S by interchanging V and W, i.e.

$$S = V^*W. \tag{8}$$

Since GMRES convergence is unitarily invariant, the pair $(W^*BW, W^*b) = (V^*W, W^*b) = (S, W^*b)$ gives also the same residual norm convergence curve. We can find a GMRES(A, b)-equivalent pair with the same unitary system matrix S but a *different* right-hand side: Using the unitary equivalence $(B, b) = (V^*WV^*V, V^*b) = (S, e_1)$ we obtain the GMRES(A, b)-equivalent pair (S, e_1). Since B is normal, we have $B = Z\Delta Z^*$ where Z is unitary and Δ is the diagonal matrix containing the eigenvalues of B. Therefore yet another GMRES(A, b)-equivalent pair is (Δ, Z^*b).

We next give a parametrization of all GMRES(A, b)-equivalent pairs with a unitary system matrix. For our result we will exploit the relationship between unitary upper Hessenberg matrices with real positive subdiagonals and the so-called Schur parameters. This relationship is briefly outlined below.

2.2.1. Unitary Hessenberg matrices and Schur parameters

Any unitary upper Hessenberg matrix of order n with positive subdiagonal entries can be uniquely parametrized by n complex parameters γ_k such that $|\gamma_k| < 1$, $k = 1, \ldots, n-1$

and $|\gamma_n| = 1$, see, e.g., [14–19] or, for a reference from functional analysis, [20, Section 4.1] where the parametrization is named *GGT representation* (after Geronimus, Gragg, and Teplzaev). We will denote such unitary upper Hessenberg matrices with Q_+ (not to be confounded with the unitary upper Hessenberg matrices Q^+ in (5) which have a positive first row but do not necessarily have a positive subdiagonal). The γ_k's are called Schur parameters (this term was introduced in [14]). They are also known as partial correlation coefficients in statistics and reflection coefficients in signal processing. It is useful to introduce the so-called complementary Schur parameters σ_k, $k = 1, \ldots, n-1$ which are real and positive such that $\sigma_k = \sqrt{1 - |\gamma_k|^2}$. The matrix Q_+ can be written as the product

$$Q_+ = G_1(\gamma_1)G_2(\gamma_2)\cdots G_{n-1}(\gamma_{n-1})\tilde{G}_n(\gamma_n),$$

where

$$G_k(\gamma_k) = \operatorname{diag}\left(I_{k-1}, \begin{bmatrix} -\gamma_k & \sigma_k \\ \sigma_k & \overline{\gamma_k} \end{bmatrix}, I_{n-k-1}\right), \qquad \tilde{G}_n(\gamma_n) = \operatorname{diag}(I_{n-1}, -\gamma_n), \tag{9}$$

and the nonzero entries of Q_+ are given by

$$q_{k+1,k} = \sigma_j, \qquad q_{j,k} = -\overline{\gamma_{j-1}}\sigma_j\sigma_{j+1}\cdots\sigma_{k-1}\gamma_k, \quad 1 \leqslant j \leqslant k. \tag{10}$$

This means that the matrix Q_+ has the following form (see [18]):

$$Q_+ = \begin{bmatrix} -\gamma_1 & -\sigma_1\gamma_2 & \cdots & \cdots & -\sigma_1\cdots\sigma_{k-1}\gamma_k & \cdots & -\sigma_1\cdots\sigma_{n-1}\gamma_n \\ \sigma_1 & -\overline{\gamma_1}\gamma_2 & \cdots & \cdots & -\overline{\gamma_1}\sigma_2\cdots\sigma_{k-1}\gamma_k & \cdots & -\overline{\gamma_1}\sigma_2\cdots\sigma_{n-1}\gamma_n \\ & \sigma_2 & -\overline{\gamma_2}\gamma_3 & \cdots & \vdots & \cdots & -\overline{\gamma_2}\sigma_3\cdots\sigma_{n-1}\gamma_n \\ & & \ddots & \ddots & & & \vdots \\ & & & \sigma_{k-1} & -\overline{\gamma_{k-1}}\gamma_k & \cdots & -\overline{\gamma_{k-1}}\sigma_k\cdots\sigma_{n-1}\gamma_n \\ & & & & & & \vdots \\ & & & & \ddots & & \\ & & & & & \sigma_{n-1} & -\overline{\gamma_{n-1}}\gamma_n \end{bmatrix}.$$

Conversely, if we know Q_+, then the Schur parameters and the complementary Schur parameters are given by

$$\gamma_k = -\frac{q_{1,k}}{\sigma_1\cdots\sigma_{k-1}}, \quad 1 \leqslant k \leqslant n, \qquad \sigma_k = q_{k+1,k}, \quad 1 \leqslant k < n, \tag{11}$$

i.e., there is a one-to-one correspondence between Schur parameters and unitary upper Hessenberg matrices with positive subdiagonal entries.

We also mention the relationship of Schur parameters with Szegö polynomials. If Q_k is the leading (in general not unitary) principal submatrix of Q_+, then

$$\psi_k(\lambda) = \det(\lambda I - Q_k),$$

is the kth Szegö polynomial for $1 \leqslant k \leqslant n$ [21, Chapter XI]. Szegö polynomials can be computed from a recurrence whose coefficients are the Schur parameters.

2.2.2. GMRES residual norms and Schur parameters

The matrices $G_k(\gamma_k)$ in (9) remind us of Givens rotations used in the standard GMRES implementation (see, e.g., [1]). The Arnoldi process applied to the pair (A, b) generates the upper Hessenberg matrix H_+ in (4) (this matrix is, in general, not unitary). Instead of the QR decomposition (5) we can consider $H_+ = Q_+\hat{R}$ where

$$Q_+^* = F_1(c_1)F_2(c_2)\cdots F_{n-1}(c_{n-1}),$$
$$F_k(c_k) = \operatorname{diag}\left(I_{k-1}, \begin{bmatrix} -\bar{c}_k & s_k \\ s_k & c_k \end{bmatrix}, I_{n-k-1}\right),$$

with Givens rotation parameters c_k and $s_k > 0$ satisfying $|c_k|^2 + |s_k|^2 = 1$. With this choice Q_+ has positive subdiagonal entries. Using (9), the Schur parameters and complementary Schur parameters of Q_+ are related to the Givens rotation parameters through

$$|\gamma_k| = |c_k|, \qquad \sigma_k = s_k, \quad k = 1, \ldots, n-1. \tag{12}$$

Moreover, it follows easily from the minimization property (1) of GMRES that, with $\|b\| = 1$,

$$\|r_k\| = \prod_{j=1}^{k} |s_j|, \quad k = 1, \ldots, n-1, \tag{13}$$

see, e.g., [2, Section 6.5.5, p. 166]. This results in the next theorem.

Theorem 7. *Consider GMRES applied to (A, b) with corresponding residual norms $\|r_0\|, \|r_1\|, \ldots, \|r_{n-1}\|$. The following assertions are equivalent:*

1. *(B, c) is GMRES(A, b)-equivalent and B is unitary.*
2. *The Arnoldi process applied to (B, c) generates the decomposition $BX = XQ_+$ where X is unitary, Q_+ is a unitary upper Hessenberg matrix with positive subdiagonal entries and the Schur parameters of Q_+ satisfy*

$$|\gamma_k| = \|r_k\|\sqrt{\frac{1}{\|r_k\|^2} - \frac{1}{\|r_{k-1}\|^2}}, \quad k = 1, \ldots, n-1.$$

Proof. For the first implication, note that Q_+ has positive subdiagonal entries because it is the Hessenberg matrix resulting from the Arnoldi process, with X denoting the

unitary matrix of the associated Arnoldi vectors. Because B is unitary by assumption, $Q_+ = X^*BX$ must be unitary, too. Also note that Q_+ is the Q factor of its own QR decomposition computed with Givens rotations that have real positive off-diagonal entries. Hence the Schur parameters γ_k and complementary Schur parameters σ_k, $k = 1, \ldots, n-1$ of Q_+ satisfy (12). Because (B, c) is GMRES(A, b)-equivalent, we have

$$\prod_{j=1}^{k} \sigma_j = \|r_k\|, \quad k = 1, \ldots, n-1,$$

see (13). A straightforward argument using induction then gives

$$\sigma_k = \frac{\|r_k\|}{\|r_{k-1}\|}, \quad k = 2, \ldots, n-1.$$

Using $\sigma_k = \sqrt{1 - |\gamma_k|^2}$ we have

$$|\gamma_k| = \|r_k\| \sqrt{\frac{1}{\|r_k\|^2} - \frac{1}{\|r_{k-1}\|^2}}.$$

For the opposite implication, first note that B is unitary because so are Q_+ and X. It follows from $|\gamma_k| = \|r_k\| \sqrt{\frac{1}{\|r_k\|^2} - \frac{1}{\|r_{k-1}\|^2}}$ and from $\sigma_k = \sqrt{1 - |\gamma_k|^2}$ that the complementary Schur parameters of Q_+ are

$$\sigma_k = \frac{\|r_k\|}{\|r_{k-1}\|}, \quad k = 2, \ldots, n-1.$$

They are identical with the Givens sines because Q_+ is the Q factor of its own QR decomposition. Then, because of (13), the kth GMRES residual norm ρ_k generated by (B, c) is

$$\rho_k = \prod_{j=1}^{k} s_j = \prod_{j=1}^{k} \sigma_j = \|r_k\|, \quad k = 1, \ldots, n-1. \qquad \square$$

We remark that with Theorem 7 we can write the entries of Q_+ as a function of the residual norms. Consider the column k of the matrix Q_+. Denoting $\gamma_k = |\gamma_k| e^{\mathbf{i}\phi_k}$, the entry in the first row is

$$q_{1,k} = -\sigma_1 \cdots \sigma_{k-1} \gamma_k = -e^{\mathbf{i}\phi_k} \left(\|r_{k-1}\|^2 - \|r_k\|^2 \right)^{1/2}. \tag{14}$$

The entry in row $j \leqslant k$ is

$$\begin{aligned} q_{j,k} &= -\gamma_{j-1} \sigma_j \cdots \sigma_{k-1} \gamma_k \\ &= -e^{\mathbf{i}(\phi_{j-1} + \phi_k)} \left(\frac{1}{\|r_{j-1}\|^2} - \frac{1}{\|r_{j-2}\|^2} \right)^{1/2} \left(\|r_{k-1}\|^2 - \|r_k\|^2 \right)^{1/2}, \end{aligned}$$

where we use the convention $\sigma_k \cdots \sigma_{k-1} \equiv 1$. Finally, as we already know, $q_{k+1,k} = \sigma_k = \|r_k\|/\|r_{k-1}\|$.

The previous theorem shows that for the upper Hessenberg matrix generated by the Arnoldi process applied to a unitary GMRES(A, b)-equivalent pair, its Schur parameters give the residual norms and, except for the phase angles, the residual norms determine the Schur parameters.

Note that the upper Hessenberg matrix analyzed in [11] is the specific matrix where all Schur parameters are chosen to be real positive. The upper Hessenberg matrix Q^+ in (5) does not have a positive subdiagonal, but the entries of its first row satisfy

$$q^+_{1,k} = \eta_k \equiv \sqrt{\|r_{k-1}\|^2 - \|r_k\|^2}, \quad k = 1, \ldots, n-1, \qquad q^+_{1,n} = \eta_n \equiv \|r_{n-1}\|,$$

see [13, Theorem 3.4], and they have the same moduli as in (14). Here η_k represents the progress GMRES makes at the iteration step k; see [4–6].

Theorem 7 leads to the following characterization.

Corollary 8. *The following assertions are equivalent:*

1. (B, c) *is GMRES*(A, b)*-equivalent and* B *is unitary.*
2. *The matrix* B *and the vector* c *are of the form*

$$B = XV^*WX^*, \qquad c = Xe_1,$$

where X *is any unitary matrix,* V *is a unitary matrix whose first* k *columns give a basis of* $\mathcal{K}_k(A, b)$ *for* $1 \leqslant k \leqslant n$ *and* W *is a unitary matrix whose first* k *columns give a basis of* $A\mathcal{K}_k(A, b)$ *for* $1 \leqslant k \leqslant n$.

Proof. With Theorem 7, (B, c) is GMRES(A, b)-equivalent and B is unitary if and only if the matrix B and the vector c are of the form

$$B = XQ_+X^*, \qquad c = Xe_1,$$

where X is unitary and Q_+ is a unitary upper Hessenberg matrix with real positive subdiagonal whose Schur parameters satisfy

$$|\gamma_k| = \|r_k\|\sqrt{\frac{1}{\|r_k\|^2} - \frac{1}{\|r_{k-1}\|^2}}, \quad k = 1, \ldots, n-1, \qquad |\gamma_n| = 1.$$

It is easy to see that all unitary Hessenberg matrices generated by unitary GMRES(A, b)-equivalent pairs are diagonal unitary row and column scalings of each other. For example, denoting $\gamma_k = |\gamma_k|e^{\mathbf{i}\phi_k}$, Q_+ is a diagonal unitary row and column scaling

$$Q_+ = D_5^* Q_{++} D_6, \quad D_5^* = \operatorname{diag}\bigl(1, e^{-\mathbf{i}\phi_1}, \ldots, e^{-\mathbf{i}\phi_{n-1}}\bigr), \; D_6 = \operatorname{diag}\bigl(e^{\mathbf{i}\phi_1}, \ldots, e^{\mathbf{i}\phi_n}\bigr) \tag{15}$$

of the upper Hessenberg matrix Q_{++} where all Schur parameters are real positive.

A particular unitary GMRES(A, b)-equivalent pair is (S, e_1) with $S = \hat{V}^*\hat{W}$ where $\hat{V}$ is a unitary matrix whose first k columns give a basis of $\mathcal{K}_k(A, b)$ and $\hat{W}$ is a unitary matrix whose first k columns give a basis of $A\mathcal{K}_k(A, b)$ for $1 \leqslant k \leqslant n$, see (8). Note that because of Lemma 3, S is a unitary row and column scaling of Q^+ in that lemma and is, in particular, upper Hessenberg. Therefore the Arnoldi process for the pair (S, e_1) generates a unitary upper Hessenberg matrix which is a diagonal unitary scaling of $\hat{V}^*\hat{W}$ and the upper Hessenberg matrix Q_+ generated by *any* unitary GMRES(A, b)-equivalent pair can be written as $Q_+ = D_7^* S D_8 = (\hat{V}D_7)^*\hat{W}D_8$ for appropriate diagonal unitary matrices D_7 and D_8. □

We see that like unitary GMRES(A, b)-equivalent *matrices* (see Theorem 5), unitary GMRES(A, b)-equivalent *pairs* are determined (here up to unitary equivalence expressed by X in Corollary 8), by orthonormal bases for the Krylov subspaces $\mathcal{K}_k(A, b)$ and Krylov residual subspaces $A\mathcal{K}_k(A, b)$ for $1 \leqslant k \leqslant n$.

2.3. Unitary spectra and convergence behavior of GMRES

It is clear from the previous sections that there exist unitary GMRES(A, b)-equivalent matrices with different spectra: With Proposition 2 the same convergence curve can be generated with the spectrum of $W\mathcal{Q}W^*$ and with the spectrum of $WD_1\mathcal{Q}W^*$ where D_1 represents any diagonal unitary scaling. Similarly, there exist unitary system matrices of GMRES(A, b)-equivalent pairs with different spectra. This follows for instance from Theorem 7, where the same convergence curve is generated for all choices of phase angles of the involved Schur parameters. We can also prove the following result.

Proposition 9. *Consider GMRES applied to an unreduced unitary Hessenberg matrix Q with the right-hand side e_1 and zero initial guess. The following assertions are equivalent:*

1. *$\tilde{Q}$ is unitary and GMRES(Q, e_1)-equivalent.*
2. *$\tilde{Q} = D_1 Q D_2$, where D_i, $i = 1, 2$ are diagonal unitary matrices.*

Proof. For all $k \leqslant n$, an orthogonal basis for $\mathcal{K}_k(Q, e_1)$ is given by the unit vectors $e_1, \ldots, e_k$ and an orthogonal basis for

$$Q\mathcal{K}_k(Q, e_1) = \operatorname{span}\{Qe_1, Q^2e_1, \ldots, Q^ke_1\}$$

is given by the first k columns of Q. Therefore, with Theorem 5, $\tilde{Q} = WV^*$ where W is a diagonal unitary column scaling of Q and V is a diagonal unitary column scaling of the identity matrix I. □

In the following numerical experiments we take $n = 50$ and for given eigenvalues on the unit circle and an initial vector, we generate the unitary upper Hessenberg matrix Q_+ with real positive subdiagonal by applying the Arnoldi process to the corresponding

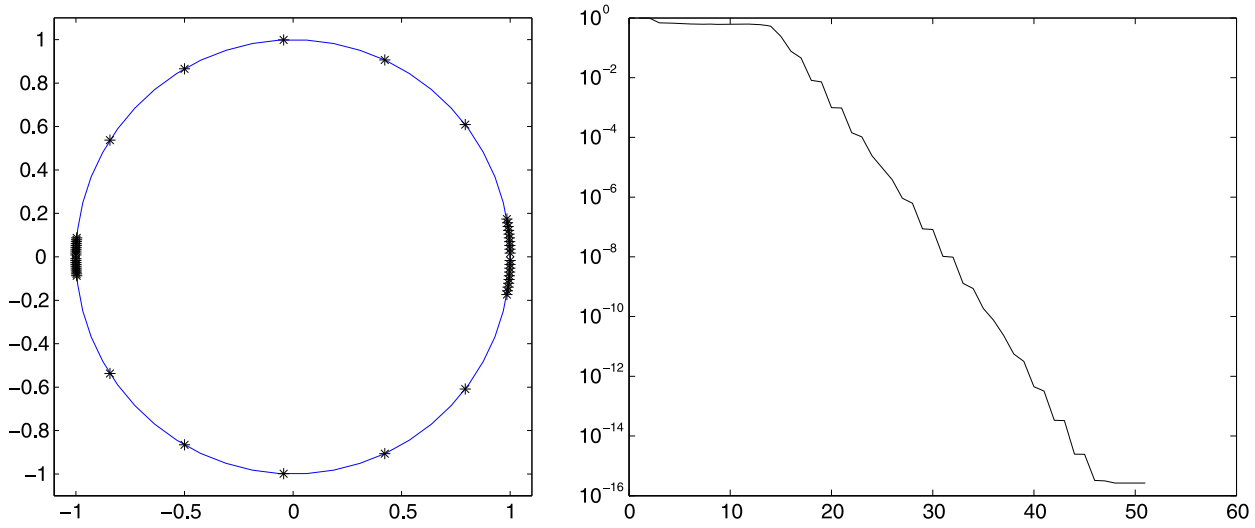


Fig. 1. Spectrum (left) of the matrix Q_+ and the GMRES residual norms for the pair (Q_+, e_1).

diagonal matrix Λ, i.e., $\Lambda X = XQ_+$. The spectrum is chosen to have two clusters within the semi-angle 10 and 5 degrees around 1 and -1 respectively, each containing 20 eigenvalues. The other 10 eigenvalues are distributed uniformly within the remaining parts of the unit circle; see the left part of Fig. 1. The initial Arnoldi vector is chosen to have all its entries equal, i.e., from $Q_+ = X^*\Lambda X$ it implies that the first column of X, which is equal to the first row of the eigenvector matrix X^* has all its entries equal to $1/\sqrt{n}$. Applying GMRES to (Q_+, e_1) gives the residual norms shown in the right part of Fig. 1. The first 12 Schur parameters of Q_+ (see (11) and Theorem 7) are of small size and the remaining ones have absolute value close to one.

Now we change the phase angles of the Schur parameters of Q_+ to make them real and positive. This gives the unitary upper Hessenberg matrix Q_{++}; see (15) and [11]. Obviously, applying GMRES to (Q_{++}, e_1) gives the same residual norms as before. The eigenvalues and the size of the squared first entries of the eigenvectors are for Q_{++}, however, different from those of Q_+. They are plotted in Fig. 2, where the eigenvectors are ordered increasingly with respect to the phase angle of the corresponding eigenvalues (with the smallest phase angle being $-\pi$ and the largest being π).

We can generate yet another GMRES(Q_+, e_1)-equivalent pair by using Proposition 9. For instance let $\tilde{Q} = D_1 Q_+ D_2$ with

$$D_1 = \operatorname{diag}\left(e^{2\pi\mathbf{i}/50}, e^{4\pi\mathbf{i}/50}, e^{6\pi\mathbf{i}/50}, \ldots, e^{2\pi\mathbf{i}}\right), \qquad D_2 = I.$$

Fig. 3 plots for $\tilde{Q}$ the information analogous to Fig. 2. The spectrum of $\tilde{Q}$ and the first components of its eigenvectors are clearly different from that of both Q_+ and Q_{++}.

Summarizing, there is no characteristic unitary spectrum corresponding to a certain GMRES convergence curve; many unitary spectra can be in general associated with the same curve. On the other hand, the bound (2) for a unitary GMRES(A, b)-equivalent pair with $\kappa(Z) = 1$ seemingly suggests a relation between a unitary spectrum and GMRES convergence. Such interpretation of (2) is, however, misleading. One must be careful with

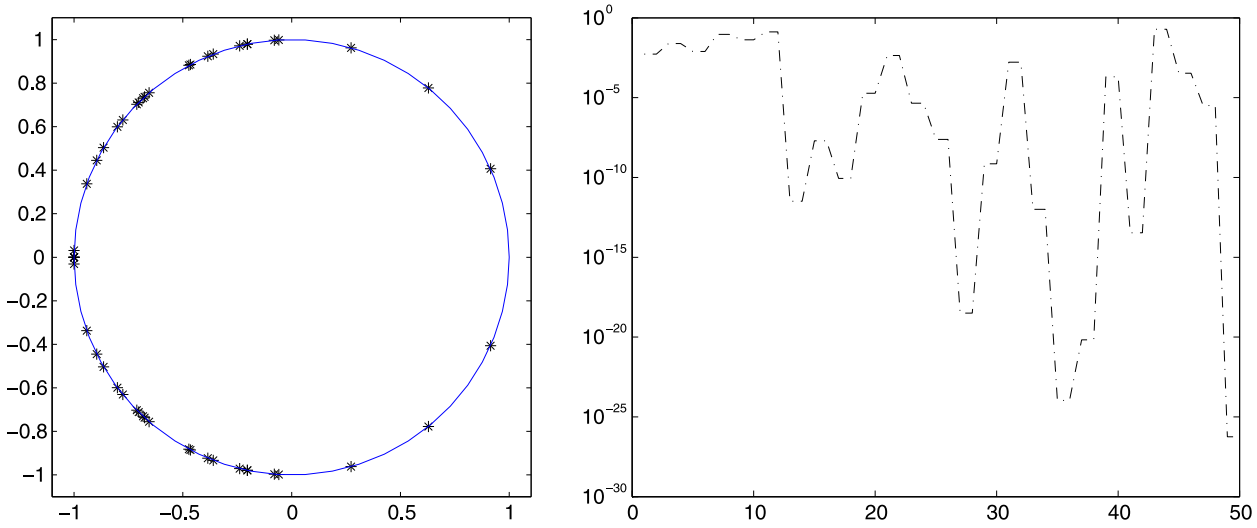


Fig. 2. Eigenvalues (left) and size of the squared first components of the associated eigenvectors of the matrix Q_{++}. The eigenvectors are in increasing order with respect to the phase angle of the eigenvalues.

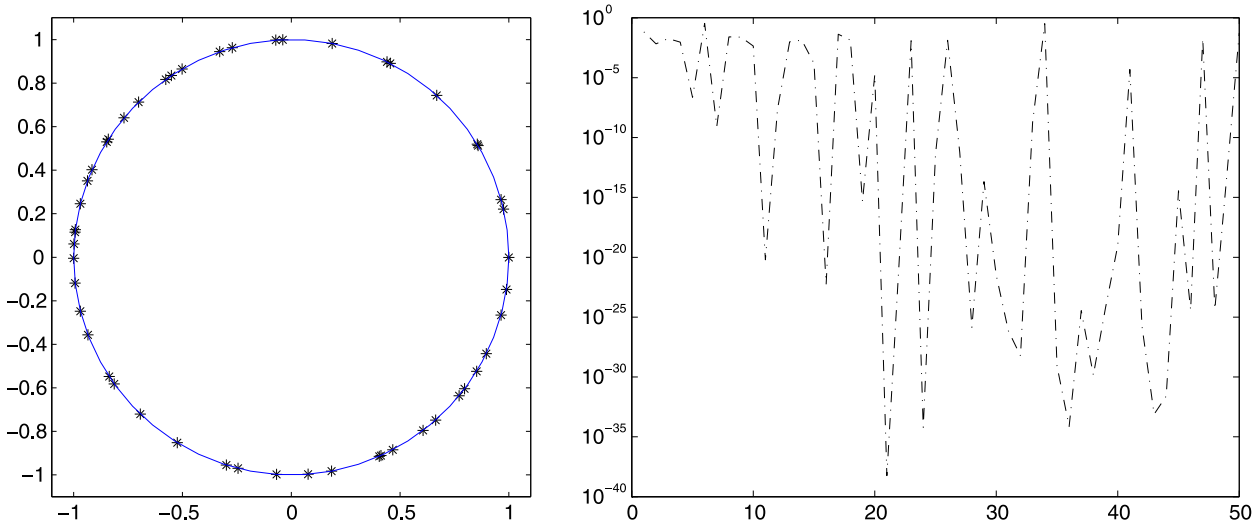


Fig. 3. Spectrum (left) and size of the squared first components of the associated eigenvectors of the matrix $\tilde{Q}$. The eigenvectors are in increasing order with respect to the phase angle of the eigenvalues.

linking the bound (2) to GMRES convergence, even if the matrix is normal. We remark that when the matrix A is Hermitian with real distinct eigenvalues, the right-hand side of (2) takes the value

$$\min_{p\in\Pi_k}\max_{i=1,\dots,n}\left|p_k(\lambda_i)\right| = \left(\sum_{j=1}^{k+1}\prod_{i=1,\,i\neq j}^{k+1}\frac{|\mu_i|}{|\mu_i-\mu_j|}\right)^{-1} \tag{16}$$

for a subset $\{\mu_1,\dots,\mu_{k+1}\}$ of $k+1$ eigenvalues of $\{\lambda_1,\dots,\lambda_n\}$, see [22]. However, as soon as one eigenvalue of A is complex, Eq. (16) does not hold in general [23]. We have the following facts for a unitary A.

If the spectrum of a unitary matrix has a large maximum gap, then we have fast GMRES convergence. This was shown in [9]. On the other hand, *if* we have fast

GMRES convergence, then there does not need to be a large gap in the spectrum of the unitary matrix A; the fast convergence can be assured by a particular decomposition of b in the invariant subspaces of A. If Q_+ is the corresponding upper Hessenberg matrix resulting from the Arnoldi process applied to (A, b), then (Q_+, e_1) is a unitary GMRES(A, b)-equivalent pair where Q_+ has the same spectrum as the unitary matrix A. Since the right-hand side e_1 for the pair (Q_+, e_1) is independent of A and b, the interplay of A and b has been in some sense wrapped into the entries of Q_+. Of course, convergence still depends on how rich e_1 is in the various eigenvectors of Q_+. However, if we scale Q_+ with unitary diagonal matrices D_1 and D_2 such that $Q_{++} = D_1 Q_+ D_2$ has only real positive Schur parameters, then, according to [11], fast convergence corresponds to a large gap in the spectrum of the Hessenberg matrix Q_{++}. The pair (Q_{++}, e_1) is another unitary GMRES(A, b)-equivalent pair (by Proposition 9). This pair does not result from the Arnoldi process for (A, b). It results from the Arnoldi process for a pair $(\tilde{A}, \tilde{b})$ where the eigenvalues of the unitary matrix $\tilde{A}$ must contain the same large gap as the spectrum of Q_{++}. GMRES applied to this matrix $\tilde{A}$ with an arbitrary right-hand side will exhibit fast convergence, see [9].

If we have stagnation of GMRES, then all corresponding unitary spectra have a very regular structure. This is shown in the following result first proved in [24]; here we give a shorter proof.

Proposition 10. *Let the GMRES method for a pair (A, b) where A has order n stagnate until the last iteration. The following assertions are equivalent:*

1. *There exists a vector c such that (B, c) is a GMRES(A, b)-equivalent pair and B is unitary.*
2. *The spectrum of the unitary matrix B is given by the roots of the equation $\lambda^n = e^{\mathbf{i}\phi}$ for a real number ϕ.*

Proof. Let (B, c) be a GMRES(A, b)-equivalent pair. Because of Theorem 7, the Arnoldi process applied to this pair generates a unitary upper Hessenberg matrix Q_+ whose Schur parameters are all zero, except for γ_n with $|\gamma_n| = 1$, i.e. $\gamma_n = e^{\mathbf{i}\phi}$ for a real number ϕ. Then the Hessenberg matrix Q_+ is nothing but the companion matrix for the polynomial $\lambda^n - e^{\mathbf{i}\phi}$ (see (10)). The claim follows because B is obtained from Q_+ using a similarity transformation, see Theorem 7. Inversely, let the spectrum of the unitary matrix B be given by the roots of the equation $\lambda^n = e^{\mathbf{i}\phi}$ for a real number ϕ and let $B = Z_B \Lambda Z_B^*$ be the spectral decomposition of B with unitary Z_B. If C is the companion matrix for the polynomial $\lambda^n - e^{\mathbf{i}\phi}$, then it is also unitary and has the spectral decomposition $C = Z_C \Lambda Z_C^*$ with unitary Z_C. We can write B as

$$B = Z_B \Lambda Z_B^* = Z_B Z_C^* C Z_C Z_B^* = X C X^*, \quad X \equiv Z_B Z_C^*,$$

with X unitary. Thus we have $BX = XC$ and if we put $X = [x_1, \ldots, x_n]$, then

$$x_j = Bx_{j-1}, \quad j = 2, \ldots, n$$

and

$$x_j = B^{j-1}x_1, \quad j = 2, \ldots, n, \qquad Bx_n = e^{\mathbf{i}\phi}x_1.$$

This means that with the choice $c \equiv x_1$

$$B\mathcal{K}_n(B, c) = \operatorname{span}\{x_2, \ldots, x_n, x_1\}.$$

Thus for $k < n$, $x_1 = c \perp B\mathcal{K}_k(B, c) = \operatorname{span}\{x_2, \ldots, x_{k+1}\}$ and GMRES applied to (B, c) stagnates until the last step. □

The previous proposition shows that complete stagnation is possible for GMRES with a unitary matrix B only if the spectrum of B represents a rotation of the roots of unity. But *if* the spectrum of a unitary matrix B represents a rotation of the roots of unity, this needs not imply complete stagnation of GMRES applied to B with an arbitrary right-hand side. It holds for some specific choice of c.

In a more general context it is worth mentioning the paper [25] where the isometric Arnoldi process is analyzed asymptotically using potential theory. In particular, the paper defines and investigates the isometric Arnoldi minimization problem and makes analogies between the properties of unitary Hessenberg matrices with positive diagonals and Jacobi matrices.

In the next section we derive an expression for the residual norms in terms of eigenvalues *and* eigenvector components when GMRES is applied to a normal matrix.

3. GMRES residual norms for normal matrices

Let the Arnoldi process applied to the pair (A, b) with normal A generate the unreduced Hessenberg matrix H_+ and the unitary matrix $\hat{V}$ satisfying (4). Since $\hat{V}^*b = e_1$, GMRES generates with (A, b) the same residual norms as with (H_+, e_1). We will therefore consider the pairs (H_+, e_1) with normal unreduced upper Hessenberg matrices. First we need the following lemma, which also holds for non-normal H_+.

Lemma 11. *Let H_+ be an unreduced Hessenberg matrix with real positive subdiagonal, C be the companion matrix corresponding to its characteristic polynomial and let*

$$U = \begin{bmatrix} e_1 & H_+e_1 & \cdots & H_+^{n-1}e_1 \end{bmatrix}, \tag{17}$$

which is an upper triangular matrix with real positive diagonal entries. Then

$$H_+ = UCU^{-1}.$$

Proof. See [26, Lemma 2] and [27, Eq. (2.4)]. □

The matrix H_+ is a *normal* unreduced upper Hessenberg matrix with real positive subdiagonal if and only if U in (17) is the Cholesky factor of a moment matrix. This is stated more precisely in the next result, which is due to Parlett [28]. Note that an unreduced normal Hessenberg matrix is diagonalizable and it has distinct eigenvalues.

Theorem 12. *Let H_+ be an unreduced Hessenberg matrix having real positive subdiagonal entries. Let all its eigenvalues λ_i, $i = 1, \ldots, n$ be distinct and let U be the upper triangular matrix in (17). Then the following statements are equivalent:*

1. *H_+ is normal.*
2. *There exist real positive weights ω_k with $\sum_{k=1}^{n} \omega_k = 1$ such that $M = U^*U$ is the moment matrix with entries defined by*

$$M_{i,j} = \sum_{k=1}^{n} \omega_k (\bar{\lambda}_k)^{i-1} \lambda_k^{j-1}. \tag{18}$$

Proof. See [28]. □

For the proof see also [27, p. 392].

The weights in the second assertion are the squares of the moduli of the first components of the eigenvectors of H_+. Indeed, with the spectral factorization $H_+ = Z \operatorname{diag}(\lambda_1, \ldots, \lambda_n) Z^* = Z \Lambda Z^*$ we get

$$U = Z \begin{bmatrix} c & \Lambda c & \cdots & \Lambda^{n-1} c \end{bmatrix}, \quad c = Z^* e_1,$$

which gives

$$M = U^*U = \begin{bmatrix} c & \Lambda c & \cdots & \Lambda^{n-1} c \end{bmatrix}^* \begin{bmatrix} c & \Lambda c & \cdots & \Lambda^{n-1} c \end{bmatrix}.$$

Comparing with (18) we obtain $\omega_k = |e_k^T c|^2$.

The residual norms generated by GMRES can be expressed in terms of the moment matrix $M = U^*U$ as follows. Let

$$K = \begin{bmatrix} b, Ab, \ldots, A^{n-1} b \end{bmatrix}$$

be the Krylov matrix for a pair (A, b). Using (4),

$$\begin{aligned} M = U^*U &= (\hat{V}U)^* \hat{V}U \\ &= \left(\hat{V} \begin{bmatrix} e_1 & H_+ e_1 & \cdots & H_+^{n-1} e_1 \end{bmatrix}\right)^* \hat{V} \begin{bmatrix} e_1 & H_+ e_1 & \cdots & H_+^{n-1} e_1 \end{bmatrix} \\ &= \left(\begin{bmatrix} b & Ab & \cdots & A^{n-1} b \end{bmatrix}\right)^* \begin{bmatrix} b & Ab & \cdots & A^{n-1} b \end{bmatrix} = K^*K. \end{aligned}$$

It has been proved in several publications (see, e.g., [29, Theorem 4.1] and [30, Lemma 1]), that if M_k denotes the kth leading principal submatrix of $M = K^*K$, then the kth GMRES residual norm $\|r_k\|$ satisfies

$$\|r_k\|^2 = \frac{1}{(M_{k+1}^{-1})_{1,1}}. \tag{19}$$

In [31, Theorem 2.1] the same result is written slightly differently and it is pointed out that the formula goes back to [32, Sections 3 and 4], see also [33, Theorem 2.1] and the remarks thereafter. Note that the formula (19) holds for general, not necessarily normal A. In the normal case it leads to the main result of this section.

Theorem 13. *Let A be a normal matrix with distinct eigenvalues and the spectral factorization $Z\Lambda Z^*$ where $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, $Z^*Z = ZZ^* = I$. Let b be a vector of unit norm such that all entries of the vector $c \equiv Z^*b$ are nonzero and let $\sum_{I_k}$ denote summation over all possible sets I_k of k indices $i_1, i_2, \ldots, i_k$ such that $1 \leqslant i_1 < \cdots < i_k \leqslant n$. The residual norms of GMRES applied to (A, b) then satisfy*

$$\|r_1\|^2 = \frac{\sum_{I_2} \omega_{i_1}\omega_{i_2} \prod_{\substack{i_1 \leqslant i_\ell < i_j \leqslant i_2 \\ i_\ell, i_j \in I_2}} |\lambda_{i_j} - \lambda_{i_\ell}|^2}{\sum_{i=1}^{n} \omega_i |\lambda_i|^2}, \tag{20}$$

and for $k = 2, \ldots, n-1$,

$$\|r_k\|^2 = \frac{\sum_{I_{k+1}} [\prod_{j=1}^{k+1} \omega_{i_j}] \prod_{\substack{i_1 \leqslant i_\ell < i_j \leqslant i_{k+1} \\ i_\ell, i_j \in I_{k+1}}} |\lambda_{i_j} - \lambda_{i_\ell}|^2}{\sum_{I_k} [\prod_{j=1}^{k} \omega_{i_j} |\lambda_{i_j}|^2] \prod_{\substack{i_1 \leqslant i_\ell < i_j \leqslant i_k \\ i_\ell, i_j \in I_k}} |\lambda_{i_j} - \lambda_{i_\ell}|^2}, \tag{21}$$

where $\omega_{i_j} = |e_{i_j}^T c|^2$.

Proof. Using (19) and Cramer's rule:

$$\|r_k\|^2 = \frac{1}{(M_{k+1}^{-1})_{1,1}} = \frac{\det(M_{k+1})}{\det(M_{2:k+1,2:k+1})},$$

where $M_{2:k+1,2:k+1}$ is the $k \times k$ trailing principal submatrix of M_{k+1}. We can write M_{k+1} as

$$M_{k+1} = \mathcal{V}_{k+1}^* D_\omega \mathcal{V}_{k+1} = \left(\mathcal{V}_{k+1}^* D_\omega^{1/2}\right)\left(D_\omega^{1/2} \mathcal{V}_{k+1}\right) \equiv F^*F, \tag{22}$$

where

$$\mathcal{V}_{k+1} = \begin{pmatrix} 1 & \lambda_1 & \cdots & \lambda_1^k \\ 1 & \lambda_2 & \cdots & \lambda_2^k \\ \vdots & \vdots & & \vdots \\ 1 & \lambda_n & \cdots & \lambda_n^k \end{pmatrix},$$

is an $n\times(k+1)$ Vandermonde matrix and D_ω a diagonal matrix of order n with $\omega_1,\dots,\omega_n$ on the diagonal. Similarly,

$$M_{2:k+1,2:k+1} = \mathcal{V}_k^* \Lambda^* D_\omega \Lambda \mathcal{V}_k = \left(\mathcal{V}_k^* D_\omega^{1/2} \Lambda^*\right)\left(\Lambda D_\omega^{1/2} \mathcal{V}_k\right) \equiv G^* G.$$

Let us first consider the determinant of M_{k+1}. Let $F_{I_{k+1},:}$ be the square submatrix of F whose row indices belong to an index set I_{k+1}. Following [34], we can use the Cauchy–Binet formula[1] for the determinant of the square product of two conforming rectangular matrices. When the rectangular matrices are Hermitian transposes of each other, the formula yields

$$\det(M_{k+1}) = \sum_{I_{k+1}} \left|\det(F_{I_{k+1},:})\right|^2.$$

Thus

$$\det(M_{k+1}) = \sum_{I_{k+1}} \left[\prod_{j=1}^{k+1} \omega_{i_j}\right] \left|\det(\mathcal{V}_{I_{k+1}})\right|^2,$$

where (see [35])

$$\mathcal{V}_{I_{k+1}} = \begin{pmatrix} 1 & \lambda_{i_1} & \cdots & \lambda_{i_1}^k \\ 1 & \lambda_{i_2} & \cdots & \lambda_{i_2}^k \\ \vdots & \vdots & & \vdots \\ 1 & \lambda_{i_{k+1}} & \cdots & \lambda_{i_{k+1}}^k \end{pmatrix}, \qquad \det(\mathcal{V}_{I_{k+1}}) = \prod_{\substack{i_1 \leqslant i_\ell < i_j \leqslant i_{k+1} \\ i_\ell, i_j \in I_{k+1}}} (\lambda_{i_j} - \lambda_{i_\ell}).$$

Analogously,

$$\begin{aligned} \det(M_{2:k+1,2:k+1}) &= \sum_{I_k} \left|\det(G_{I_k,:})\right|^2 \\ &= \sum_{I_k} \left[\prod_{j=1}^{k} \omega_{i_j} |\lambda_{i_j}|^2\right] \left|\det(\mathcal{V}_{I_k})\right|^2. \end{aligned}$$

Noting that for $k=1$, the matrix $\mathcal{V}_{I_k}$ reduces to the number one, we have

$$\sum_{I_1} \left[\prod_{j=1}^{1} \omega_{i_j} |\lambda_{i_j}|^2\right] \left|\det(\mathcal{V}_{I_1})\right|^2 = \sum_{i=1}^{n} \omega_i |\lambda_i|^2 \left|\det(1)\right|^2,$$

which finishes the proof. □

Assuming that GMRES applied to a normal matrix A and a given right-hand side vector b does not terminate until the last step n, this theorem gives the GMRES residual

[1] This formula was first proved in 1812 independently by Augustin-Louis Cauchy (1789–1857) and Jacques Binet (1786–1856).

norms in terms of the eigenvalues and the squared size of the components of the right-hand side vector in the direction of the individual eigenvectors. Thus Theorem 13 gives the solution of the polynomial approximation problem (3).

It can easily be extended to the case where GMRES terminates before the step n. If A has $m < n$ distinct eigenvalues and b has nonzero components in all m associated invariant subspaces, then GMRES terminates with $r_m = 0$, (20) holds and if $m > 2$, (21) holds for $k = 2, \ldots, m-1$. If b has nonzero components only in $\ell < m$ invariant subspaces corresponding to distinct eigenvalues, then GMRES terminates with $r_\ell = 0$, (20) holds and if $\ell > 2$, (21) holds for $k = 2, \ldots, \ell - 1$.

It should be pointed out that Ipsen gave in [31, Theorem 4.1] another expression for $\|r_k\|$ using a minimization problem over $k+1$ eigenvalues. In [23, Theorem 2.1], the formula (21) was derived for $k = n-1$. Formulas (20) and (21) might be of use in situations where the influence of the right-hand side is of interest. For instance, restarting in GMRES corresponds to changes in the weights ω_{i_j}. Worst case behavior corresponds to taking the maximum over the values of ω_{i_j}. Theorem 13 also leads to a straightforward lower and upper bound where the influence of b is separated from the influence of the spectrum.

Corollary 14. *With the notation of Theorem 13, let*

$$\omega_- = \min_{1 \leqslant i \leqslant n} \omega_i, \qquad \omega_+ = \max_{1 \leqslant i \leqslant n} \omega_i.$$

Then the residual norms of GMRES applied to (A, b) satisfy

$$\frac{\sum_{I_2} \prod_{\substack{i_1 \leqslant i_\ell < i_j \leqslant i_2 \\ i_\ell, i_j \in I_2}} |\lambda_{i_j} - \lambda_{i_\ell}|^2}{\sum_{i=1}^n |\lambda_i|^2} \omega_- \leqslant \|r_1\|^2 \leqslant \frac{\sum_{I_2} \prod_{\substack{i_1 \leqslant i_\ell < i_j \leqslant i_2 \\ i_\ell, i_j \in I_2}} |\lambda_{i_j} - \lambda_{i_\ell}|^2}{\sum_{i=1}^n |\lambda_i|^2} \omega_+, \tag{23}$$

and for $k = 2, \ldots, n-1$,

$$\|r_k\|^2 \geqslant \frac{\sum_{I_{k+1}} \prod_{\substack{i_1 \leqslant i_\ell < i_j \leqslant i_{k+1} \\ i_\ell, i_j \in I_{k+1}}} |\lambda_{i_j} - \lambda_{i_\ell}|^2}{\sum_{I_k} [\prod_{j=1}^k |\lambda_{i_j}|^2] \prod_{\substack{i_1 \leqslant i_\ell < i_j \leqslant i_k \\ i_\ell, i_j \in I_k}} |\lambda_{i_j} - \lambda_{i_\ell}|^2} \omega_-, \tag{24}$$

$$\|r_k\|^2 \leqslant \frac{\sum_{I_{k+1}} \prod_{\substack{i_1 \leqslant i_\ell < i_j \leqslant i_{k+1} \\ i_\ell, i_j \in I_{k+1}}} |\lambda_{i_j} - \lambda_{i_\ell}|^2}{\sum_{I_k} [\prod_{j=1}^k |\lambda_{i_j}|^2] \prod_{\substack{i_1 \leqslant i_\ell < i_j \leqslant i_k \\ i_\ell, i_j \in I_k}} |\lambda_{i_j} - \lambda_{i_\ell}|^2} \omega_+. \tag{25}$$

Proof. If C and D are two matrices of sizes $n \times (k+1)$ and $n \times n$ respectively, $k \leqslant n-1$, and C is of full rank, then

$$\frac{\sigma_{min}(D)^2}{e_1^T (C^* C)^{-1} e_1} \leqslant \frac{1}{e_1^T (C^* (D^* D) C)^{-1} e_1} \leqslant \frac{\sigma_{max}(D)^2}{e_1^T (C^* C)^{-1} e_1}, \tag{26}$$

see [36, Lemma 1]. If we put, using the notation of the proof of Theorem 13, $C \equiv \mathcal{V}_{k+1}$ and $D \equiv D_\omega^{1/2}$, then the claim follows using exactly the same arguments as in the proof of Theorem 13. □

The bounds are attained for $\omega_- = \omega_+$, that is, for all components of the right hand side in the eigenvector basis of the same size. In that case the bounds are attained for all k, i.e. one choice of b guarantees the equality throughout the entire GMRES process. In comparison, the standard bound (2) with $\kappa(Z) = 1$ is fully dependent on eigenvalues, but it is attained at iteration k for a specific right-hand side which depends upon k. Moreover, (2) is an upper bound only; Corollary 14 gives both lower and upper bounds showing the descriptive role of eigenvalues for the behavior of GMRES in the normal case.

When A is unitary, the eigenvalues are of modulus one and (20) and (21) simplify to

$$\|r_1\|^2 = \sum_{I_2} \omega_{i_1}\omega_{i_2} \prod_{\substack{i_1 \leqslant i_\ell < i_j \leqslant i_2 \\ i_\ell, i_j \in I_2}} |\lambda_{i_j} - \lambda_{i_\ell}|^2$$

and

$$\|r_k\|^2 = \frac{\sum_{I_{k+1}} \prod_{j=1}^{k+1} \omega_{i_j} \prod_{\substack{i_1 \leqslant i_\ell < i_j \leqslant i_{k+1} \\ i_\ell, i_j \in I_{k+1}}} |\lambda_{i_j} - \lambda_{i_\ell}|^2}{\sum_{I_k} \prod_{j=1}^{k} \omega_{i_j} \prod_{\substack{i_1 \leqslant i_\ell < i_j \leqslant i_k \\ i_\ell, i_j \in I_k}} |\lambda_{i_j} - \lambda_{i_\ell}|^2}, \tag{27}$$

respectively. We see that the GMRES convergence for unitary matrices depends strongly on the angles between pairs of eigenvalues on the unit circle. As it is obvious, the residual norms stay the same when all eigenvalues are rotated by a given angle (without modifying the eigenvectors).

Formula (27) for the GMRES residual norm, which holds at every iteration step, offers an insight to the fact that outlying eigenvalues can often be associated with an initial stage of slow GMRES convergence (see, e.g., [37] where the emphasis is on the asymptotic convergence factor after the initial stage). From (27) we see that if there is one tight cluster of eigenvalues and, say, m other eigenvalues well separated from this cluster, then after $m + 1$ iterations there will be at least one small factor $|\lambda_{i_j} - \lambda_{i_\ell}|^2$ in every summation term of the numerator because $m + 2$ eigenvalues are involved and at least one pair of eigenvalues will belong to the cluster. If the weights are of similar size, one can therefore expect acceleration of convergence after the m initial steps.

An analogous argument can be used in the presence of multiple clusters; with ℓ well separated clusters and another m well separated single eigenvalues, acceleration might be expected after $m + \ell$ steps. This offers an explanation for the acceleration of convergence after 13 steps in Fig. 1, Section 2.3, where the spectrum was chosen to have two clusters, the other 10 eigenvalues were regularly distributed around the remaining portions of the unit circle and all weights were chosen to be equal. Here the sharpness of

the start of the acceleration as well as the approximate slope of the convergence curve for $k > 13$ depend on the tightness of the clusters in comparison with the mutual distance of the well-separated eigenvalues. A bad configuration is when the eigenvalues are almost regularly distributed on the unit circle and the weights are all of similar size.

Corollary 8 and (27) finally give the following statement.

Theorem 15. *Let V be a unitary matrix whose first k columns give a basis of the Krylov space $\mathcal{K}_k(A,b)$ and W be a unitary matrix whose first k columns give a basis of the Krylov residual space $A\mathcal{K}_k(A,b)$ for $1 \leqslant k \leqslant n$. Using the notation of Theorem 13, the residual norms of GMRES applied to (A,b) satisfy*

$$\|r_1\|^2 = \sum_{I_2} \omega_{i_1}\omega_{i_2} \prod_{\substack{i_1 \leqslant i_\ell < i_j \leqslant i_2 \\ i_\ell, i_j \in I_2}} |\delta_{i_j} - \delta_{i_\ell}|^2,$$

$$\|r_k\|^2 = \frac{\sum_{I_{k+1}} \prod_{j=1}^{k+1} \omega_{i_j} \prod_{\substack{i_1 \leqslant i_\ell < i_j \leqslant i_{k+1} \\ i_\ell, i_j \in I_{k+1}}} |\delta_{i_j} - \delta_{i_\ell}|^2}{\sum_{I_k} \prod_{j=1}^{k} \omega_{i_j} \prod_{\substack{i_1 \leqslant i_\ell < i_j \leqslant i_k \\ i_\ell, i_j \in I_k}} |\delta_{i_j} - \delta_{i_\ell}|^2}, \quad k = 2, \ldots, n,$$

where the δ_i are the eigenvalues in the generalized eigenvalue problem

$$Wx = \delta Vx$$

and the ω_i are the squared moduli of the first components of the corresponding eigenvectors.

4. Conclusion

We investigate GMRES(A,b)-equivalent pairs (B,c), where B is unitary. We characterize B in terms of orthonormal bases for the sequence of Krylov subspaces $\mathcal{K}_k(A,b)$ and Krylov residual subspaces $A\mathcal{K}_k(A,b)$, $k = 1,2,\ldots,n$. This shows that a possible linking of the spectral properties of unitary GMRES(A,b)-equivalent matrices, which influence GMRES convergence behavior, to some simple properties of A would be, in general, rather difficult. We also offer some insight concerning the substantial role of the right-hand side vector components in the direction of the individual eigenvectors. The presented formula giving the residual norms for normal matrices can for some particular eigenvalue distribution explain acceleration of convergence of GMRES observed after a number of iterations.

Acknowledgements

We thank the referees for the very insightful reports, for many helpful comments which clarified the presentation and for pointing out Refs. [20,27,37]. We thank Heike

Faßbender for pointing out Ref. [14]. The work of J. Duintjer Tebbens was supported by the institutional support RVO:67985807 and by the project M100301201 of the Grant Agency of the ASCR. The work of Z. Strakoš has been supported by the ERC-CZ project LL1202.

References

[1] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 7 (3) (1986) 856–869.
[2] Y. Saad, Iterative Methods for Sparse Linear Systems, second ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.
[3] A. Greenbaum, L.N. Trefethen, GMRES/CR and Arnoldi/Lanczos as matrix approximation problems, SIAM J. Sci. Comput. 15 (2) (1994) 359–368.
[4] A. Greenbaum, Z. Strakoš, Matrices that generate the same Krylov residual spaces, in: Recent Advances in Iterative Methods, in: IMA Vol. Math. Appl., vol. 60, Springer, New York, 1994, pp. 95–118.
[5] A. Greenbaum, V. Pták, Z. Strakoš, Any nonincreasing convergence curve is possible for GMRES, SIAM J. Matrix Anal. Appl. 17 (3) (1996) 465–469.
[6] M. Arioli, V. Pták, Z. Strakoš, Krylov sequences of maximal length and convergence of GMRES, BIT 38 (4) (1998) 636–643.
[7] J. Liesen, Z. Strakoš, Krylov Subspace Methods, Principles and Analysis, Oxford University Press, 2012.
[8] J. Duintjer Tebbens, G. Meurant, Any Ritz value behavior is possible for Arnoldi and for GMRES, SIAM J. Matrix Anal. Appl. 33 (3) (2012) 958–978.
[9] J. Liesen, Computable convergence bounds for GMRES, SIAM J. Matrix Anal. Appl. 21 (3) (2000) 882–903.
[10] J. Liesen, Construction and analysis of polynomial iterative methods for non-hermitian systems of linear equations, PhD thesis, University of Bielefeld, Germany, 1998.
[11] L. Knizhnerman, On GMRES-equivalent bounded operators, SIAM J. Matrix Anal. Appl. 22 (1) (2000) 195–212.
[12] H.F. Walker, L. Zhou, A simpler GMRES, Numer. Linear Algebra Appl. 1 (6) (1994) 571–581.
[13] G. Meurant, GMRES and the Arioli, Pták, and Strakoš parametrization, BIT 52 (3) (2012) 687–702.
[14] W. Gragg, The QR algorithm for unitary Hessenberg matrices, J. Comput. Appl. Math. 16 (1986) 1–8.
[15] G.S. Ammar, W.B. Gragg, L. Reichel, Constructing a unitary Hessenberg matrix from spectral data, in: G.H. Golub, P. Van Dooren (Eds.), Numerical Linear Algebra, Digital Signal Processing, and Parallel Algorithms, Springer, 1991, pp. 385–396.
[16] A. Bunse-Gerstner, L. Elsner, Schur parameter pencils for the solution of the unitary eigenproblem, Linear Algebra Appl. 154–156 (1991) 741–778.
[17] G.S. Ammar, C.Y. He, On an inverse eigenvalue problem for unitary Hessenberg matrices, Linear Algebra Appl. 218 (1995) 263–271.
[18] H. Faßbender, Inverse unitary eigenproblems and related orthogonal functions, Numer. Math. 77 (1997) 323–345.
[19] T.-L. Wang, W.B. Gragg, Convergence of the shifted QR algorithm for unitary Hessenberg matrices, Math. Comp. 71 (240) (2002) 1473–1496.
[20] B. Simon, Orthogonal Polynomials on the Unit Circle. Part 1, Amer. Math. Soc. Colloq. Publ., vol. 54, American Mathematical Society, Providence, RI, 2005.
[21] G. Szegő, Orthogonal Polynomials, fourth ed., Amer. Math. Soc. Colloq. Publ., vol. XXIII, American Mathematical Society, Providence, RI, 1975.
[22] A. Greenbaum, Comparison of splittings used with the conjugate gradient algorithm, Numer. Math. 33 (2) (1979) 181–193.
[23] J. Liesen, P. Tichý, The worst-case GMRES for normal matrices, BIT 44 (1) (2004) 79–98.
[24] I. Zavorin, D.P. O'Leary, H. Elman, Complete stagnation of GMRES, Linear Algebra Appl. 367 (2003) 165–183.
[25] S. Helsen, A.B.J. Kuijlaars, M. Van Barel, Convergence of the isometric Arnoldi process, SIAM J. Matrix Anal. Appl. 26 (3) (2005) 782–809.

[26] B. Parlett, Canonical decomposition of Hessenberg matrices, Math. Comp. 21 (98) (1967) 223–227.
[27] A. Ruhe, Rational Krylov sequence methods for eigenvalue computation, Linear Algebra Appl. 58 (1984) 391–405.
[28] B. Parlett, Normal Hessenberg and moment matrices, Linear Algebra Appl. 6 (1973) 37–43.
[29] J. Zítko, Generalization of convergence conditions for a restarted GMRES, Numer. Linear Algebra Appl. 7 (2000) 117–131.
[30] H. Sadok, Analysis of the convergence of the minimal and the orthogonal residual methods, Numer. Algorithms 40 (2) (2005) 201–216.
[31] I.C.F. Ipsen, Expressions and bounds for the GMRES residual, BIT 40 (3) (2000) 524–535.
[32] G.W. Stewart, Collinearity and least squares regression, Statist. Sci. 2 (1) (1987) 68–100.
[33] J. Liesen, M. Rozložník, Z. Strakoš, Least squares residuals and minimal residual methods, SIAM J. Sci. Comput. 23 (5) (2002) 1503–1525.
[34] M. Bellalij, H. Sadok, A new approach to GMRES convergence, 2011, unpublished report.
[35] W. Gautschi, On inverses of Vandermonde and confluent Vandermonde matrices. III, Numer. Math. 29 (1977/1978) 445–450.
[36] M. Bellalij, K. Jbilou, H. Sadok, New convergence results on the global GMRES method for diagonalizable matrices, J. Comput. Appl. Math. 219 (2) (2008) 350–358.
[37] S.L. Campbell, I.C.F. Ipsen, C.T. Kelley, C.D. Meyer, GMRES and the minimal polynomial, BIT 36 (4) (1996) 664–675.

Numer Algor
DOI 10.1007/s11075-014-9891-3

ORIGINAL PAPER

The role eigenvalues play in forming GMRES residual norms with non-normal matrices

Gérard Meurant · Jurjen Duintjer Tebbens

Received: 21 April 2014 / Accepted: 30 June 2014

Abstract In this paper we give explicit expressions for the norms of the residual vectors generated by the GMRES algorithm applied to a non-normal matrix. They involve the right-hand side of the linear system, the eigenvalues, the eigenvectors and, in the non-diagonalizable case, the principal vectors. They give a complete description of how eigenvalues contribute in forming residual norms and offer insight in what quantities can prevent GMRES from being governed by eigenvalues.

Keywords GMRES convergence · Non-normal matrix · Eigenvalues · Residual norms

1 Introduction

We consider the convergence of GMRES (the Generalized Minimal RESidual method) for solving linear systems with complex nonsingular matrices A of size n and n-dimensional right-hand sides b; see e.g. [38] or [37] for a description of the algorithm. The kth GMRES iterate x_k minimizes, with $x_0 = 0$, the norm of the kth residual vector $r_k = b - Ax_k$ over all vectors in the kth Krylov subspace

G. Meurant
30 rue du sergent Bauchat, 75012 Paris, France
e-mail: gerard.meurant@gmail.com

J. Duintjer Tebbens (✉)
Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 2, 18 207 Praha 8, Libeň, Prague, Czech Republic
e-mail: duintjertebbens@cs.cas.cz

J. Duintjer Tebbens
Faculty of Pharmacy in Hradec Králové, Charles University in Prague, Heyrovského 1203, 500 05 Hradec Králové, Czech Republic

Published online: 16 July 2014

$\mathcal{K}_k(A, b) \equiv \text{span}\{b,\ Ab,\ \ldots,\ A^{k-1}b\}$. Therefore, residual norms are non-increasing and satisfy

$$\|r_k\| = \min_{p \in \pi_k} \|p(A)b\|,$$

where π_k is the set of polynomials of degree k with the value one at the origin and $\|\cdot\|$ denotes the 2-norm. If the Jordan canonical form of A is denoted by $A = XJX^{-1}$, then

$$\|r_k\| = \min_{p \in \pi_k} \|X\, p\,(J) X^{-1} b\|. \tag{1}$$

In this paper we focus on how convergence of the GMRES residual norms is influenced by the entirety of spectral properties of A, that is, by the eigenvalues contained in J and by the eigenvectors or principal vectors contained in X.

If A is Hermitian, the orthogonality of the eigenvectors results in a predominant influence of the eigenvalues on convergence. For example, in Hermitian counterparts of GMRES like the MINRES method [34] or the Conjugate Gradients method [19], clustering of eigenvalues stimulates convergence, eigenvalues close to zero hamper convergence and the eigenvalue distribution decides about the rate of convergence (for a survey, see, e.g., [27]). In addition, there exist for these methods sharp upper bounds consisting of a min-max problem which depends on the spectrum only. For instance, in MINRES the residual norms satisfy

$$\frac{\|r_k\|}{\|b\|} \leq \min_{p \in \pi_k} \max_{i=1,\ldots,n} |p_k(\lambda_i)|, \tag{2}$$

with λ_i denoting the eigenvalues of A (see, e.g., [37]) and for every k there exists a right-hand side (depending on k) such that equality holds. MINRES is a method for Hermitian matrices which is mathematically equivalent with GMRES, thus the residual norms generated by GMRES applied to a Hermitian matrix satisfy the same inequality. In fact, it is satisfied with normal matrices too, and in this case, GMRES convergence is governed by eigenvalues as well. Moreover, from (1) we have for any normal matrix

$$\|r_k\| = \min_{p \in \pi_k} \|p\,(J) X^* b\|, \tag{3}$$

with J being a diagonal matrix of eigenvalues. This shows that with Hermitian or other normal matrices, the residual norms are fully determined by two quantities: eigenvalues and components of the right-hand side in the eigenvector basis. A closed-form expression for the kth GMRES residual norm in terms of these quantities (in fact of the moduli of the components of the right-hand side in the eigenvector basis), i.e. the solution of (3), was presented in [10] and in an unpublished report from Bellalij and Sadok (A new approach to GMRES convergence, 2011).

When A is not normal, the predominant role of the eigenvalues can be lost. For diagonalizable non-normal matrices, the upper bound (2) is multiplied with the condition number $\kappa(X)$ of the eigenvector matrix, which may be large. We refer to [26, Section 3.1] for a detailed discussion of other difficulties with interpreting this bound in the non-normal case. The probably most convincing results showing that GMRES need not be governed only by eigenvalues can be found in a series of papers by Arioli, Greenbaum, Pták and Strakoš [1, 17, 18]. They show that for any prescribed sequence of n non-increasing residual norms, there exists a class of

right-hand sides and matrices, whose nonzero eigenvalues can be chosen *arbitrarily*, giving residual norms that coincide with the given non-increasing sequence. In this sense, GMRES convergence curves (with respect to residual norms) are independent from the eigenvalues of A. It was shown in [8] that convergence curves do not even depend on the Ritz values generated during all iterations of the GMRES process. The strong potential independence from eigenvalues inspired many papers that look for some approaches other than eigenvalue analysis to explain GMRES convergence. They include pseudospectra [33, 44], the field of values [11], the polynomial numerical hull [16], potential theory [23], decomposition in normal plus low-rank [20] or comparison with GMRES for non-Euclidean inner products [36]. Though they can be very suited to explain convergence for particular problems, none of the approaches seems to represent a universal tool for GMRES analysis.

Nevertheless for many practical problems, eigenvalues seem to influence convergence behavior strongly. This follows for instance from the fact that slow convergence can often be successfully cured by eliminating particular convergence hampering eigenvalues with a so-called deflation strategy; see, to mention just some of a large number of proposed techniques, for instance [2, 5–7, 12, 14, 15, 22, 24, 29–32, 35]. This is not surprising since residual vectors are formed from a matrix polynomial times the right-hand side and matrix polynomials are naturally related to eigenvalues. It is often assumed that the situation where the behavior of GMRES is not or little governed by eigenvalues occurs only for matrices that are far from normal. However, even such a highly non-normal matrix as a Jordan block can yield GMRES convergence curves that are dominated by the size of the involved eigenvalue (this will also be discussed in Section 3 of this paper). In fact, Arioli, Greenbaum, Pták and Strakoš never wrote in [1, 17, 18] that GMRES convergence does not depend on the eigenvalues. The results in [1, 17, 18] merely show that there are sets of matrices with different (arbitrary) eigenvalue distributions and right-hand sides giving the same GMRES residual norms. In view of (1) this means that if one modifies eigenvalues, then in order to have the same residual norms, the eigenvectors and/or principal vectors and the right-hand side must and can be modified appropriately.

In this paper we address the interplay of eigenvalues, eigenvectors and the right-hand side with respect to convergence. In the first place, our goal is to show as precisely as possible, how eigenvalues contribute to the computation of residual norms. To this end, we derive closed-form expressions for the residual norms. In the second place, we use these expressions in an attempt to enhance insight in when convergence can be suspected to be dominated by the spectrum and when not. We discuss several interpretations of departure from normality, the role of the right-hand side and the frequently observed convergence hampering influence of eigenvalues close to the origin. For ease of presentation we will not consider the early termination case in detail, though in practice, of course, one often terminates the process after a small number of iterations. With early termination we obtain the same closed-form expressions but for a smaller number of iterations and this leads to exactly the same insights.

The contents of the paper are as follows. In Section 2 we give an expression of the GMRES residual norms for diagonalizable matrices. Section 3 generalizes the

ideas of the previous section for matrices with one Jordan block and Section 4 treats the more general case when the matrix A is not diagonalizable. We formulate some conclusions in the last section. Throughout the paper we will use the phrase "convergence is governed by eigenvalues" when convergence depends only on eigenvalues and on components of the right-hand side in the eigenvector basis; eigenvectors and right-hand side do not influence convergence curves in any other way. This is the case for GMRES applied to normal matrices, see (3), for the MINRES method, and, with respect to the norm of the A-error, the Conjugate Gradients method. We will assume that GMRES does not terminate before iteration n. Hence, the Krylov subspaces are of full dimension and their orthogonal bases constructed using the Gram-Schmidt algorithm are well defined. For the sake of simplicity we choose $x_0 = 0$ and we normalize the right-hand side b such that $\|r_0\| = \|b\| = 1$. The vector e_i will denote the ith column of the identity matrix (of appropriate order). The entry on the ith row and in the jth column of a matrix X is denoted by $X_{i,j}$ and $X_{i:j,k:\ell}$ denotes the submatrix of X with rows from i to j and columns from k to ℓ. $X_{i:j,:}$ denotes the submatrix with rows from i to j and with all columns of X.

2 GMRES convergence for diagonalizable matrices

In this section we look for the solution of the minimization problem (1) in terms of J, X and $X^{-1}b$ when A is diagonalizable with spectral factorization $X\Lambda X^{-1}$ where the eigenvalues are contained in $\Lambda = J = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$. To this end, we generalize the results in [10] and in the unpublished report from Bellalij and Sadok (A new approach to GMRES convergence, 2011) that solved the minimization problem (3) for normal matrices. The next sections will address the non-diagonalizable case.

Let

$$K = \begin{pmatrix} b & Ab & A^2b & \cdots & A^{n-1}b \end{pmatrix},$$

be the Krylov matrix whose first k columns are the natural basis vectors of the Krylov subspace $\mathcal{K}_k(A, b)$ for $1 \leq k \leq n$ and let $c = X^{-1}b$. Then the Krylov matrix K can be written as $K = X \begin{pmatrix} c & \Lambda c & \cdots & \Lambda^{n-1}c \end{pmatrix}$ and let us define the moment matrix.

$$M = K^*K = \begin{pmatrix} c & \Lambda c & \cdots & \Lambda^{n-1}c \end{pmatrix}^* X^*X \begin{pmatrix} c & \Lambda c & \cdots & \Lambda^{n-1}c \end{pmatrix} \tag{4}$$

For all Krylov subspaces to have full dimension we need the eigenvalues to be distinct and c to have no zero entries. We remark that it is easily seen from the parametrizations in [1] and [9] that any non-increasing GMRES convergence curve is possible for diagonalizable matrices with any distinct eigenvalues. We now try to show how eigenvectors and components of the right-hand side must be modified if we wish to generate the same residual norms with different distinct eigenvalues.

The residual norms in GMRES are given by

$$\|r_k\|^2 = \frac{1}{e_1^T M_{k+1}^{-1} e_1}, \qquad k = 1, \ldots, n-1, \tag{5}$$

where M_{k+1} is the leading principal submatrix of order $k+1$ of M. This result has been proved independently in several papers; see [45, Theorem 4.1], [21, Theorem

2.1] where the result is formulated differently using a pseudo-inverse and [39, Lemma 1] where it is given for real matrices. In [25, Theorem 2.1] and the remarks thereafter it is pointed out that the formula goes back to [40, Section 3 and 4]. As in [10] and in the unpublished report from Bellalij and Sadok (A new approach to GMRES convergence, 2011), the (1, 1) entry of M_{k+1}^{-1} in (5) will be calculated using Cramer's rule:

$$(M_{k+1}^{-1})_{1,1} = \frac{\det(M_{2:k+1,2:k+1})}{\det(M_{k+1})}. \tag{6}$$

With D_c denoting the diagonal matrix whose diagonal entries c_i are the components of c and with

$$\mathcal{V}_{k+1} = \begin{pmatrix} 1 & \lambda_1 & \cdots & \lambda_1^k \\ 1 & \lambda_2 & \cdots & \lambda_2^k \\ \vdots & \vdots & & \vdots \\ 1 & \lambda_n & \cdots & \lambda_n^k \end{pmatrix}, \tag{7}$$

an $n \times (k+1)$ matrix, we see that M_{k+1} in (6) can be written as

$$M_{k+1} = \mathcal{V}_{k+1}^* D_c^* X^* X D_c \mathcal{V}_{k+1}. \tag{8}$$

If $F \equiv X D_c \mathcal{V}_{k+1}$, then M_{k+1} is the product F^*F of two rectangular matrices. To compute the determinants of M_{k+1} and $M_{2:k+1,2:k+1}$ in (6) we will use the Cauchy-Binet formula for determinants of products of rectangular matrices: For the product of a $(k \times n)$ matrix G with an $(n \times k)$ matrix H there holds

$$\det(GH) = \sum_{I_k} \det(G_{:,I_k}) \det(H_{I_k,:}).$$

The notation used here is clear from the following definitions, which we will need in the sequel.

Definition 1 With I_k (or J_k) we denote sets of k ordered indices $i_1, \ldots, i_k$ such that $1 \le i_1 < \cdots < i_k \le n$. With $\sum_{I_k}$ we denote summation over all such possible ordered index sets. With X_{I_k,J_k} we denote the square $k \times k$ submatrix of X whose row and column indices of entries are defined respectively by I_k and J_k. With $\prod_{j_\ell < j_p \in J_k}$ we denote the product over all pairs of indices j_ℓ, j_p in the ordered index set J_k such that $j_\ell < j_p$.

Having outlined the main proof ingredients, we now give the resulting expressions of the residual norm for GMRES processes that do not terminate before iteration n. We remark that they can be used for the case where GMRES terminates before the step n as follows: If A has $m < n$ distinct eigenvalues and b has nonzero components in all m associated invariant subspaces, then GMRES terminates with $r_m = 0$, and the expressions presented below hold for $k = 1, \ldots, m-1$. If b has nonzero components only in $\ell < m$ invariant subspaces corresponding to distinct eigenvalues, then GMRES terminates with $r_\ell = 0$ and the expressions holds for $k = 1$ and if $\ell > 2$, for $k = 2, \ldots, \ell - 1$.

The next theorem does not contain very elegant formulaes, but it gives the solution of (1) in the case where J is a diagonal matrix.

Theorem 1 *Let A be a diagonalizable matrix with a spectral factorization $X\Lambda X^{-1}$ where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ contains the distinct eigenvalues and let b be a vector of unit norm such that $c = X^{-1}b$ has no zero entries. When solving $Ax = b$ with $x_0 = 0$, the GMRES residual norm at iteration $k < n$ satisfies*

$$\|r_k\|^2 = \sigma_{k+1}^N / \sigma_k^D,$$

where

$$\sigma_{k+1}^N = \sum_{I_{k+1}} \left| \sum_{J_{k+1}} \det(X_{I_{k+1},J_{k+1}})\, c_{j_1} \cdots c_{j_{k+1}} \prod_{j_\ell < j_p \in J_{k+1}} (\lambda_{j_p} - \lambda_{j_\ell}) \right|^2,$$

$\sigma_1^D = \sum_{i=1}^n \left| \sum_{j=1}^n X_{i,j}\, c_j\, \lambda_j \right|^2$, *and for* $k \geq 2$

$$\sigma_k^D = \sum_{I_k} \left| \sum_{J_k} \det(X_{I_k,J_k})\, c_{j_1} \cdots c_{j_k}\, \lambda_{j_1} \cdots \lambda_{j_k} \prod_{j_\ell < j_p \in J_k} (\lambda_{j_p} - \lambda_{j_\ell}) \right|^2.$$

Proof We apply Cramer's rule (6) to compute the (1, 1) entry of the inverse of M_{k+1}. Let us first consider the determinant of M_{k+1}. By the Cauchy-Binet formula,

$$\det(M_{k+1}) = \sum_{I_{k+1}} |\det(F_{I_{k+1},:})|^2.$$

Thus we have to compute the determinant of $F_{I_{k+1},:}$, a matrix which consists of rows $i_1, \ldots, i_{k+1}$ of $XD_c\mathcal{V}_{k+1}$. It is the product of a $(k+1) \times n$ matrix that we can write as $(XD_c)_{I_{k+1},:}$ by the $n \times (k+1)$ matrix $\mathcal{V}_{k+1}$. Once again we can use the Cauchy-Binet formula. Let

$$\mathcal{V}(\lambda_{j_1}, \ldots, \lambda_{j_{k+1}}) = \begin{pmatrix} 1 & \lambda_{j_1} & \cdots & \lambda_{j_1}^k \\ 1 & \lambda_{j_2} & \cdots & \lambda_{j_2}^k \\ \vdots & \vdots & & \vdots \\ 1 & \lambda_{j_{k+1}} & \cdots & \lambda_{j_{k+1}}^k \end{pmatrix}$$

which is a square Vandermonde matrix of order $k+1$. Then

$$\det(F_{I_{k+1},:}) = \sum_{J_{k+1}} \det(X_{I_{k+1},J_{k+1}}) c_{j_1} \cdots c_{j_{k+1}} \det(\mathcal{V}(\lambda_{j_1}, \ldots, \lambda_{j_{k+1}})).$$

Moreover, we have (see, e.g. [13])

$$\det(\mathcal{V}(\lambda_{j_1}, \ldots, \lambda_{j_{k+1}})) = \prod_{j_\ell < j_p \in J_{k+1}} (\lambda_{j_p} - \lambda_{j_\ell}).$$

Finally, the determinant of M_{k+1} is

$$\sigma_{k+1}^N = \sum_{I_{k+1}} \left| \sum_{J_{k+1}} \det(X_{I_{k+1},J_{k+1}}) c_{j_1} \cdots c_{j_{k+1}} \prod_{j_\ell < j_p \in J_{k+1}} (\lambda_{j_p} - \lambda_{j_\ell}) \right|^2.$$

Let us now consider the determinant of $M_{2:k+1,2:k+1}$ which is a matrix of order k. The computation is essentially the same, except that we have to consider the rows

and columns 2 to $k+1$. Therefore, it is not $\mathcal{V}_k$ which is involved any longer but a matrix that can be written as $\Lambda\mathcal{V}_k$. We have

$$M_{2:k+1,2:k+1} = \mathcal{V}_k^* \Lambda^* D_c^* X^* X D_c \Lambda \mathcal{V}_k.$$

Then, we have some additional factors arising from the diagonal matrix Λ and we have to consider only sets of k indices I_k and J_k. The determinant of $M_{2:k+1,2:k+1}$ is obtained, for $k > 1$, as

$$\sigma_k^D = \sum_{I_k} \left| \sum_{J_k} \det(X_{I_k,J_k}) c_{j_1} \cdots c_{j_k} \lambda_{j_1} \cdots \lambda_{j_k} \prod_{j_\ell < j_p \le J_k} (\lambda_{j_p} - \lambda_{j_\ell}) \right|^2 .$$

Noting that for $k = 1$, the matrix $\mathcal{V}_{I_k}$ reduces to the number one, we have

$$\begin{aligned} \sigma_1^D &= \sum_{I_1} \sum_{J_1} \left| \det(X_{I_1,J_1}) c_{j_1} \cdots c_{j_1} \lambda_{j_1} \cdots \lambda_{j_1} \det(\mathcal{V}_{I_1}) \right|^2 \\ &= \sum_{i=1}^{n} \left| \sum_{j=1}^{n} X_{i,j}\, c_j\, \lambda_j \det(1) \right|^2 . \end{aligned}$$

The residual norm squared is finally given as $\|r_k\|^2 = \sigma_{k+1}^N / \sigma_k^D$. □

Theorem 1 shows in what manner the norm of the residual vector depends on the eigenvalues (through eigenvalue products and products of eigenvalue differences), on the eigenvectors (through determinants of submatrices of the eigenvector matrix) and on $c = X^{-1}b$ (through products of its entries). Theorem 1 seems to support the frequently observed fact that eigenvalues close to the origin tend to hamper convergence. The common explanation for this behavior is that it is difficult for GMRES to construct, when it terminates, a polynomial with the value one in the origin which is zero in an eigenvalue close to zero. Theorem 1 shows that, with diagonalizable matrices, a spectrum close to the origin may cause many terms in the denominators σ_k^D to be close to zero and may give relatively large residual norms. Of course, the papers [1, 17, 18] proved that small eigenvalues need not hamper convergence in general.

As we mentioned in the introduction, a standard upper bound for GMRES residual norms with diagonalizable matrices is

$$\frac{\|r_k\|}{\|b\|} \le \kappa(X) \min_{p \in \pi_k} \max_{i=1,\ldots,n} |p_k(\lambda_i)|, \tag{9}$$

see, e.g., [38]. This bound suggests that the condition number $\kappa(X)$ of the eigenvector matrix plays an important role for convergence behavior. But according to Theorem 1, GMRES residual norms are not explicitly dependent on $\kappa(X)$. The eigenvector matrix X has a large impact, but its inverse is present only through the entries of $c = X^{-1}b$ (which is also clear from (1)). With an appropriate right-hand side, the influence of a large value of $\|X^{-1}\|$ can be eliminated and give a vector c with entries of moderate size.

When the matrix A is normal, we have $X^*X = I$ and the sums over J_k and J_{k+1} reduce to only one term ($J_k = I_k$, respectively $J_{k+1} = I_{k+1}$). We then recover

the formula in [10] and in the unpublished report from Bellalij and Sadok (A new approach to GMRES convergence, 2011).

Theorem 2 *Let A be a normal matrix with distinct eigenvalues and the spectral factorization $X\Lambda X^*$ where $\Lambda = \text{diag}\,(\lambda_1, \ldots, \lambda_n)$, $X^*X = XX^* = I$. Let b be a vector of unit norm such that all entries of the vector $c = X^*b$ are nonzero. When solving $Ax = b$ with $x_0 = 0$, the GMRES residual norm at iteration $k = 1$ satisfies*

$$\|r_1\|^2 = \frac{\sum_{I_2} \omega_{i_1}\omega_{i_2} \prod_{i_\ell < i_j \in I_2} |\lambda_{i_j} - \lambda_{i_\ell}|^2}{\sum_{i=1}^{n} \omega_i |\lambda_i|^2}, \tag{10}$$

and for $k = 2, \ldots, n-1$,

$$\|r_k\|^2 = \frac{\sum_{I_{k+1}} \left[\prod_{j=1}^{k+1} \omega_{i_j}\right] \prod_{i_\ell < i_j \in I_{k+1}} |\lambda_{i_j} - \lambda_{i_\ell}|^2}{\sum_{I_k} \left[\prod_{j=1}^{k} \omega_{i_j} |\lambda_{i_j}|^2\right] \prod_{i_\ell < i_j \in I_k} |\lambda_{i_j} - \lambda_{i_\ell}|^2}, \tag{11}$$

where $\omega_{i_j} = |e_{i_j}^T c|^2$.

We remark that equations (10) and (11) were derived in [28, Theorem 2.1] for $k = n-1$ and that the equations (for all k) hold as well for the residual norms generated by the mathematically equivalent MINRES method for Hermitian (and so normal) matrices.

When A is normal, GMRES residual norms depend on the eigenvectors and the right-hand side *only* through the sizes ω_i of the squared components of the right-hand side in the eigenvector basis (which is also clear from (3)). Therefore, the role of eigenvalues is much more pronounced than in the non-normal case. If A is close to normal in the sense that $X^*X \approx I$, then in the numerators σ_{k+1}^N and denominators σ_k^D of Theorem 1 the involved determinants of submatrices of X may be small except for the choices $J_{k+1} = I_{k+1}$, respectively $J_k = I_k$, but this has to be investigated further. We can, however, derive bounds from Theorem 1 that involve the conditioning of X. We derive them with the help of the following bounds that can be found in [3].

Lemma 1 *Let G and H be two matrices of sizes $n \times (k+1)$ and $n \times n$ respectively, $k \leq n-1$. If the matrix G is of full rank,*

$$\frac{\sigma_{min}(H)^2}{e_1^T (G^*G)^{-1} e_1} \leq \frac{1}{e_1^T (G^*(H^*H)G)^{-1} e_1} \leq \frac{\sigma_{max}(H)^2}{e_1^T (G^*G)^{-1} e_1}. \tag{12}$$

Proposition 1 *Let A be a matrix with distinct eigenvalues and the spectral factorization $X\Lambda X^{-1}$ where $\Lambda = \text{diag}\,(\lambda_1, \ldots, \lambda_n)$. Let b be a vector of unit norm such that all entries of the vector $c \equiv X^{-1}b$ are nonzero. When solving $Ax = b$ with $x_0 = 0$,*

the GMRES residual norm at iteration $k = 1$ satisfies

$$\|r_1\|^2 \geq \sigma_{min}(X)^2 \frac{\sum_{I_2} \omega_{i_1}\omega_{i_2} \prod_{i_\ell<i_j\in I_2} |\lambda_{i_j} - \lambda_{i_\ell}|^2}{\sum_{i=1}^n \omega_i |\lambda_i|^2},$$
$$\|r_1\|^2 \leq \|X\|^2 \frac{\sum_{I_2} \omega_{i_1}\omega_{i_2} \prod_{i_\ell<i_j\in I_2} |\lambda_{i_j} - \lambda_{i_\ell}|^2}{\sum_{i=1}^n \omega_i |\lambda_i|^2},$$

and for $k = 2, \ldots, n-1$,

$$\|r_k\|^2 \geq \sigma_{min}(X)^2 \frac{\sum_{I_{k+1}} \left[\prod_{j=1}^{k+1} \omega_{i_j}\right] \prod_{i_\ell<i_j\in I_{k+1}} |\lambda_{i_j} - \lambda_{i_\ell}|^2}{\sum_{I_k} \left[\prod_{j=1}^{k} \omega_{i_j} |\lambda_{i_j}|^2\right] \prod_{i_\ell<i_j\in I_k} |\lambda_{i_j} - \lambda_{i_\ell}|^2},$$
$$\|r_k\|^2 \leq \|X\|^2 \frac{\sum_{I_{k+1}} \left[\prod_{j=1}^{k+1} \omega_{i_j}\right] \prod_{i_\ell<i_j\in I_{k+1}} |\lambda_{i_j} - \lambda_{i_\ell}|^2}{\sum_{I_k} \left[\prod_{j=1}^{k} \omega_{i_j} |\lambda_{i_j}|^2\right] \prod_{i_\ell<i_j\in I_k} |\lambda_{i_j} - \lambda_{i_\ell}|^2},$$

where $\omega_{i_j} = |e_{i_j}^T c|^2$.

Proof Because of (5) and (8), we have

$$\|r_k\|^2 = \frac{1}{e_1^T (\mathcal{V}_{k+1}^* D_c^* (X^* X) D_c \mathcal{V}_{k+1})^{-1} e_1}.$$

Applying Lemma 1 with $G \equiv D_c \mathcal{V}_{k+1}$ and $H \equiv X$ we obtain

$$\frac{\sigma_{min}(X)^2}{e_1^T (\mathcal{V}_{k+1}^* D_c^* D_c \mathcal{V}_{k+1})^{-1} e_1} \leq \|r_k\|^2 \leq \frac{\|X\|^2}{e_1^T (\mathcal{V}_{k+1}^* D_c^* D_c \mathcal{V}_{k+1})^{-1} e_1}.$$

The claim follows by realizing that the value $1/e_1^T (\mathcal{V}_{k+1}^* D_c^* D_c \mathcal{V}_{k+1})^{-1} e_1$ is precisely the squared residual norm for a linear system with normal matrix having eigenvalues $\lambda_1, \ldots, \lambda_n$ and such that $c = X^{-1} b$. □

The bounds in the previous proposition are attained if $\kappa(X) = 1$ and are in some sense a two-sided alternative to (9). They show that if $\sigma_{min}(X)$ is close to $\sigma_{max}(X)$, then residual norms behave essentially as in the normal case and are governed by eigenvalues. However, the opposite need not be true. If $\kappa(X)$ is large, the question whether convergence is dominated by the spectrum of A will depend on the interplay with the entries of $c = X^{-1} b$ and determinants of X. If we wish to derive bounds similar to those in Proposition 1 where the eigenvalues are fully separated from eigenvectors and right-hand side, this can be done as follows.

Proposition 2 *Let A be a matrix with distinct eigenvalues and the spectral factorization $X \Lambda X^{-1}$ where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$. Let b be a vector of unit norm such that all entries of the vector $c \equiv X^{-1} b$ are nonzero and let D_c denote the diagonal*

matrix whose diagonal entries c_i are the components of c. When solving $Ax = b$ with $x_0 = 0$, the GMRES residual norm at iteration $k = 1$ satisfies

$$\|r_1\|^2 \geq \sigma_{min}(XD_c)^2 \frac{\sum_{I_2} \prod_{i_\ell < i_j \in I_2} |\lambda_{i_j} - \lambda_{i_\ell}|^2}{\sum_{i=1}^n |\lambda_i|^2},$$

$$\|r_1\|^2 \leq \|XD_c\|^2 \frac{\sum_{I_2} \prod_{i_\ell < i_j \in I_2} |\lambda_{i_j} - \lambda_{i_\ell}|^2}{\sum_{i=1}^n |\lambda_i|^2},$$

and for $k = 2, \ldots, n-1$,

$$\|r_k\|^2 \geq \sigma_{min}(XD_c)^2 \frac{\sum_{I_{k+1}} \prod_{i_\ell < i_j \in I_{k+1}} |\lambda_{i_j} - \lambda_{i_\ell}|^2}{\sum_{I_k} \left[\prod_{j=1}^k |\lambda_{i_j}|^2\right] \prod_{i_\ell < i_j \in I_k} |\lambda_{i_j} - \lambda_{i_\ell}|^2},$$

$$\|r_k\|^2 \leq \|XD_c\|^2 \frac{\sum_{I_{k+1}} \prod_{i_\ell < i_j \in I_{k+1}} |\lambda_{i_j} - \lambda_{i_\ell}|^2}{\sum_{I_k} \left[\prod_{j=1}^k |\lambda_{i_j}|^2\right] \prod_{i_\ell < i_j \in I_k} |\lambda_{i_j} - \lambda_{i_\ell}|^2}.$$

Proof Because of (5) and (8), we have

$$\|r_k\|^2 = \frac{1}{e_1^T (\mathcal{V}_{k+1}^* D_c^* (X^* X) D_c \mathcal{V}_{k+1})^{-1} e_1}.$$

Applying Lemma 1 with $G \equiv \mathcal{V}_{k+1}$ and $H \equiv XD_c$ we obtain

$$\frac{\sigma_{min}(XD_c)^2}{e_1^T (\mathcal{V}_{k+1}^* \mathcal{V}_{k+1})^{-1} e_1} \leq \|r_k\|^2 \leq \frac{\|XD_c\|^2}{e_1^T (\mathcal{V}_{k+1}^* \mathcal{V}_{k+1})^{-1} e_1}.$$

The claim follows in the same way as in the proof of Proposition 1. □

The bounds in this proposition may be tight even if the condition number of the eigenvector matrix X is large: $D_c = \mathrm{diag}(c)$ may represent a favorable scaling of the eigenvector matrix. In fact, as D_c contains X^{-1} through $c = X^{-1}b$, in some particular cases the influence of X^{-1} in the product XD_c might be cancelled out by X. For other bounds that incorporate the right-hand side through $X^{-1}b$ we refer to [43], where the scaling of X is also discussed.

Because for diagonalizable matrices, "departure from normality" can be translated to "size of the condition number of the eigenvector matrix", we conclude that GMRES for diagonalizable matrices close to normal will be governed by the spectrum. With a more important departure from normality, the degree to which eigenvalues govern GMRES will depend upon the interplay with determinants of X and entries of $X^{-1}b$; even with a high condition number $\kappa(X)$, GMRES behavior can be governed by the spectrum in particular cases.

3 One Jordan block

We start our investigation of how Theorem 1 can be extended to the non-diagonalizable case by considering the situation where the Jordan canonical form

of A has one Jordan block only. Let A have the Jordan form XJX^{-1} with $J = \text{bidiag}(\lambda, 1)$ for a nonzero eigenvalue λ and let b be a vector of unit norm such that the last entry of $c = X^{-1}b$ is nonzero (otherwise GMRES terminates before the nth iteration). Then the moment matrix M is

$$M = K^*K = \begin{pmatrix} c & Jc & \cdots & J^{n-1}c \end{pmatrix}^* X^*X \begin{pmatrix} c & Jc & \cdots & J^{n-1}c \end{pmatrix}.$$

In contrast with the Krylov matrix $\begin{pmatrix} c & \Lambda c & \cdots & \Lambda^{n-1}c \end{pmatrix} = D_c\mathcal{V}_n$ in the previous section (see (4) and (7)), the Krylov matrix $\begin{pmatrix} c & Jc & \cdots & J^{n-1}c \end{pmatrix}$ cannot be written as the product of a diagonal matrix containing the entries of c with a Vandermonde matrix. Instead, it can be decomposed as

$$\begin{pmatrix} c & Jc & \cdots & J^{n-1}c \end{pmatrix} = CE \equiv \tag{13}$$

$$\begin{pmatrix} c_1 & c_2 & \ldots & \ldots & c_n \\ & c_2 & c_3 & \ldots & c_n \\ & & c_3 & \ldots & c_n \\ & & \vdots & & c_n \\ & & c_n & & \end{pmatrix} \begin{pmatrix} 1 & \lambda & \lambda^2 & \cdots & \lambda^{n-1} \\ 0 & 1 & 2\lambda & \cdots & \binom{n-1}{1}\lambda^{n-2} \\ 0 & 0 & 1 & \cdots & \binom{-1}{2}\lambda^{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix},$$

where the matrix C is a Hänkel "anti upper triangular" matrix defined by $c_1, \ldots, c_n, 0, \cdots, 0$. Here is a small example for illustration: Let $n = 5$ and let all entries of $c = X^{-1}b$ be nonzero. Then the Krylov matrix $\begin{pmatrix} c & Jc & \cdots & J^4c \end{pmatrix}$ is

$$\begin{pmatrix} c_1 & \lambda c_1 + c_2 & \lambda^2 c_1 + 2\lambda c_2 + c_3 & \lambda^3 c_1 + 3\lambda^2 c_2 + 3\lambda c_3 + c_4 & \lambda^4 c_1 + 4\lambda^3 c_2 + 6\lambda^2 c_3 + 4\lambda c_4 + c_5 \\ c_2 & \lambda c_2 + c_3 & \lambda^2 c_2 + 2\lambda c_3 + c_4 & \lambda^3 c_2 + 3\lambda^2 c_3 + 3\lambda c_4 + c_5 & \lambda^4 c_2 + 4\lambda^3 c_3 + 6\lambda^2 c_4 + 4\lambda c_5 \\ c_3 & \lambda c_3 + c_4 & \lambda^2 c_3 + 2\lambda c_4 + c_5 & \lambda^3 c_3 + 3\lambda^2 c_4 + 3\lambda c_5 & \lambda^4 c_3 + 4\lambda^3 c_4 + 6\lambda^2 c_5 \\ c_4 & \lambda c_4 + c_5 & \lambda^2 c_4 + 2\lambda c_5 & \lambda^3 c_4 + 3\lambda^2 c_5 & \lambda^4 c_4 + 4\lambda^3 c_5 \\ c_5 & \lambda c_5 & \lambda^2 c_5 & \lambda^3 c_5 & \lambda^4 c_5 \end{pmatrix}$$

with the factorization

$$\begin{pmatrix} c & Jc & \cdots & J^4c \end{pmatrix} = \begin{pmatrix} c_1 & c_2 & c_3 & c_4 & c_5 \\ & c_2 & c_3 & c_4 & c_5 \\ & & c_3 & c_4 & c_5 \\ & & & c_4 & c_5 \\ & & & & c_5 \end{pmatrix} \begin{pmatrix} 1 & \lambda & \lambda^2 & \lambda^3 & \lambda^4 \\ & 1 & 2\lambda & 3\lambda^2 & 4\lambda^3 \\ & & 1 & 3\lambda & 6\lambda^2 \\ & & & 1 & 4\lambda \\ & & & & 1 \end{pmatrix}.$$

The $(k+1)$st leading principal submatrix M_{k+1} of M is given by

$$M_{k+1} = \begin{pmatrix} c & Jc & \cdots & J^kc \end{pmatrix}^* X^*X \begin{pmatrix} c & Jc & \cdots & J^kc \end{pmatrix}.$$

With (13) and defining

$$Y \equiv XC,$$

we have

$$M_{k+1} = (E_{:,1:k+1})^*(XC)^*XCE_{:,1:k+1} = (E_{:,1:k+1})^*Y^*YE_{:,1:k+1},$$

which can be written as the product $M_{k+1} = F^*F$ of two rectangular matrices where $F \equiv YE_{:,1:k+1}$. The matrix $E_{:,1:k+1}$ depends only on the eigenvalue, the matrix Y

contains all information from the principal vectors and the right-hand side. Using exactly the same proof technique as for Theorem 1, we obtain for a single Jordan block the following.

Corollary 1 *Let A be a nonsingular matrix with a single eigenvalue λ and with Jordan form XJX^{-1} where $J = \text{bidiag}(\lambda, 1)$. Let b be a vector of unit norm such that the last entry of $c = X^{-1}b$ is nonzero, let E be the eigenvalue matrix defined by (13) and let $Y = XC$, where C is the Hänkel matrix defined in (13). When solving $Ax = b$ with $x_0 = 0$, the GMRES residual norm at iteration $k < n$ satisfies*

$$\|r_k\|^2 = \frac{\sum_{I_{k+1}} \left| \sum_{J_{k+1}} \det(Y_{I_{k+1},J_{k+1}}) \det(E_{J_{k+1},1:k+1}) \right|^2}{\sum_{I_k} \left| \sum_{J_k} \det(Y_{I_k,J_k}) \det(E_{J_k,2:k+1}) \right|^2}. \tag{14}$$

Corollary 1 shows an interplay between eigenvalues, principal vectors and right-hand side which is similar to the interplay between eigenvalues, eigenvectors and right-hand side in Theorem 1. GMRES residual norms are formed from polynomials in the eigenvalue on the one hand and from determinants of the principal vector matrix multiplied with a matrix containing the entries of $X^{-1}b$ on the other hand. The inverse X^{-1} of the matrix of principal vectors X appears only in combination with the right-hand side through the vector $c = X^{-1}b$ and as before, possible ill-conditioning of X does not necessarily have a significant influence on convergence behavior.

One can prove an analogue of Proposition 1 by applying Lemma 1 with $G \equiv CE$ and $H \equiv X$. It would show that if $\kappa(X) = 1$, the behavior of GMRES applied to a very defective matrix is still governed by the eigenvalue, i.e. influenced only by the spectrum and the components of b in X (in particular c_n may be important). This would correspond to the special and somewhat superficial situation where A has a single Jordan block and where the matrix X is unitary, i.e. the Jordan form of A is $A = XJX^*$. For example, GMRES for a single, plain Jordan block is, in general, strongly governed by the eigenvalue (see, e.g., the results for a single Jordan block in [26] and [42]). Matrices of the form $A = XJX^*$ are far from normal in the sense of being maximally defective. Clearly, this type of departure from normality of A does not decide upon whether GMRES is governed by eigenvalues. As in the previous section, the departure from *orthogonality* of the eigenvector or principal vectors tells us something. If $\kappa(X)$ is large, the degree to which the spectrum governs convergence behavior is influenced by the entries of X and $c = X^{-1}b$ (an analogue of Proposition 2 for one Jordan block is possible too).

Some simplifications of the expression (14) are given by the next lemmas. The numerator of $\|r_k\|^2$ contains the determinants of $E_{J_{k+1},1:k+1}$ for all index sets J_{k+1}. Their values are given in the following result.

Lemma 2 *For all the sets of $k+1$ indices J_{k+1} in the numerator of (14), the only determinant of $E_{J_{k+1},1:k+1}$ which is non-zero is $\det(E_{1:k+1,1:k+1}) = 1$.*

Proof We have to consider all the sets of indices j_ℓ such that $1 \le j_1 < \cdots < j_{k+1} \le n$. Since E is upper triangular, all the determinants involving a row of index larger

than $k+1$ are zero. The only set of indices J_{k+1} without a row of index larger than $k+1$ is $\{1, 2, \ldots, k+1\}$. The corresponding submatrix is triangular with ones on the diagonal. □

From Lemma 2 there is only one term for the sum over J_{k+1} in the numerator σ_{k+1}^N in (14) and

$$\sigma_{k+1}^N = \sum_{I_{k+1}} \left|\det(Y_{I_{k+1},1:k+1})\right|^2 .$$

We remark that in this case the numerator does not depend on the eigenvalue. For the denominator in (14) we are interested in the determinants of $E_{J_k,2:k+1}$. They are characterized in the following lemma.

Lemma 3 *The k+1 non-zero determinants of $E_{J_k,2:k+1}$ are obtained for the sets of indices J_k not containing an index strictly larger than $k+1$. If those sets are enumerated in lexicographic order, the determinants are respectively $\lambda^k, \lambda^{k-1}, \ldots, \lambda, 1$. Moreover, the denominator σ_k^D for $\|r_k\|^2$ in (14) is*

$$\sigma_k^D = \sum_{I_k} \left|\lambda^k \det(Y_{I_k,\mathcal{I}_1}) + \quad \cdots \quad + \lambda \det(Y_{I_k,\mathcal{I}_k}) + \det(Y_{I_k,\mathcal{I}_{k+1}})\right|^2 ,$$

where $\mathcal{I}_j$, $j = 1, \ldots, k+1$, are the sets of indices with k elements in the ordered combinations of $k+1$ elements enumerated in lexicographic ordering.

Proof The first claim is obvious since if there is a row index strictly larger than $k+1$ in J_k then there is a zero row in the matrix $E_{J_k,2:k+1}$ and the determinant is zero. The proof of the second claim is by induction on k. For $k = 1$ the only nonzero determinants of $E_{J_1,2}$ are, in lexicographical order, $\det(E_{1,2}) = E_{1,2} = \lambda$ and $\det(E_{2,2}) = E_{2,2} = 1$. Let us assume that the claim is true for $k-1$. We have to consider the determinants of submatrices of order k of the $n \times k$ matrix

$$E_{:,2:k+1} = \begin{pmatrix} \lambda & \lambda^2 & \cdots & \lambda^{k-1} & \lambda^k \\ 1 & 2\lambda & \cdots & \binom{k-1}{1}\lambda^{k-2} & \binom{k}{1}\lambda^{k-1} \\ 0 & 1 & \cdots & \binom{k-1}{2}\lambda^{k-3} & \binom{k}{2}\lambda^{k-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & \binom{k}{k-1}\lambda \\ 0 & 0 & \cdots & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix} .$$

In lexicographic order the first set of indices J_k is $\{1, 2, \ldots, k\}$. We have to consider the determinant of the matrix $E^{(k)}$ obtained from the first k rows of $E_{:,2:k+1}$. Let us compute this determinant using the last column. It is equal to

$$(-1)^{k+1}[\lambda^k \det(E^{(k)}_{-1,1:k-1}) - \binom{k}{1}\lambda^{k-1}\det(E^{(k)}_{-2,1:k-1}) - \cdots + (-1)^{k-1}\binom{k}{k-1}\lambda \det(E^{(k)}_{-k,1:k-1})],$$

where $\det(E^{(k)}_{-j,1:k-1})$ denotes the determinant of the square submatrix of order $k-1$ of $E^{(k)}$ from columns 1 to $k-1$ with row j removed. Those determinants are given by our induction hypothesis (in reverse order); they are $1, \lambda, \ldots, \lambda^{k-1}$. Therefore we can factor λ^k in the expression displayed above and we obtain

$$(-1)^{k+1}\lambda^k[1 - \binom{k}{1} + \binom{k}{2} - \cdots + (-1)^{k-1}\binom{k}{k-1}].$$

One can see that the sum within brackets is equal to $(-1)^{k+1}$ and thus the determinant we were looking for is λ^k. The proof for the other sets of indices J_k is along the same lines. □

Combining Lemmas 3 and 2 with Corollary 1, we obtain the next theorem. Note that if the given right-hand side is sparse this may influence the nonzero pattern of Y and cause the annihilation of some further determinants.

Theorem 3 *Let A be a nonsingular matrix with a single eigenvalue λ and with Jordan form XJX^{-1} where $J = \text{bidiag}(\lambda, 1)$. Let b be a vector of unit norm such that the last entry of $c = X^{-1}b$ is nonzero, let E be the eigenvalue matrix defined by (13) and let $Y = XC$, where C is as defined in (13). When solving $Ax = b$ with $x_0 = 0$, the GMRES residual norm at iteration $k < n$ satisfies*

$$\|r_k\|^2 = \frac{\sum_{I_{k+1}} |\det(Y_{I_{k+1},1:k+1})|^2}{\sum_{I_k} |\lambda^k \det(Y_{I_k,\mathcal{I}_1}) + \cdots + \lambda \det(Y_{I_k,\mathcal{I}_k}) + \det(Y_{I_k,\mathcal{I}_{k+1}})|^2}, \qquad (15)$$

where $\mathcal{I}_j$, $j = 1, \ldots, k+1$ are the sets of indices with k elements in the ordered combinations of $k+1$ elements enumerated in lexicographic ordering.

Another result for the residual norms generated by GMRES applied to a Jordan block was given in [21]. The expression in that paper contains constants whose values are generally unknown.

We observe from Theorem 3 an interesting, slightly enhanced independence from the spectrum in comparison with diagonalizable matrices: The numerator is fully independent from the eigenvalue and so are the summands $\det(Y_{I_k,\mathcal{I}_{k+1}})$ in the denominator. In the expression for residual norms of Theorem 1 all summands in both numerator and denominator depend on eigenvalues.

We next consider a very small convection-diffusion model problem where matrices close to a single Jordan block arise. We also examine the corresponding Jordan block for which the theory holds exactly. The choice of the number of inner nodes for

discretization and of the source term are physically somewhat artificial but we made these choices for the sake of showing that the formulae for the residual norm can be useful.

Consider the one-dimensional convection-diffusion problem on the unit interval [0, 1]

$$-\nu u'' + u' = f, \qquad u(0) = u(1) = 0,$$

discretized with finite differences on a regular grid with n inner nodes using upwind differences for the convective term. This gives a linear system where the system matrix A is tridiagonal with entries

$$A = h^{-2}\,\mathrm{tridiag}(-\nu - h, 2\nu + h, -\nu),$$

see, e.g. [41, Section 4]. In the convection dominated case, $\nu \ll h^2$ and A is close to a scaled transposed Jordan block. Let the source term be nonzero only around the first inner node $1/(n+1)$, with the value $(\nu + h)/(-h^2)$ in that node. Then the right-hand side b is a multiple of e_1 and GMRES applied to the pair (A, b) gives the same residual norms as GMRES applied to the pair

$$\left(\frac{-h^2}{\nu + h} I^- A I^-, \frac{-h^2}{\nu + h} I^- b\right), \tag{16}$$

where I^- denotes the (unitary) antidiagonal reversion matrix with ones on the antidiagonal. The matrix $\frac{-h^2}{\nu+h} I^- A I^-$ is a near Jordan block with the eigenvalue $\lambda = -(2\nu + h)/(\nu + h)$, the right-hand side is e_n.

In the left part of Fig. 1 we show the GMRES residual norms generated with the pair (16), where $n = 4$ and $\nu = 0.01$ (dashed lines). We also show the convergence curve for the same pair, except that the lower subdiagonal entries of A have been put to zero to obtain a true Jordan block (dotted lines). Clearly, the convergence of GMRES applied to the pair (A, b) is very close to that for a Jordan block with eigenvalue $\lambda = -(2\nu + h)/(\nu + h) = -1.0476$ and right-hand side e_n. Below we

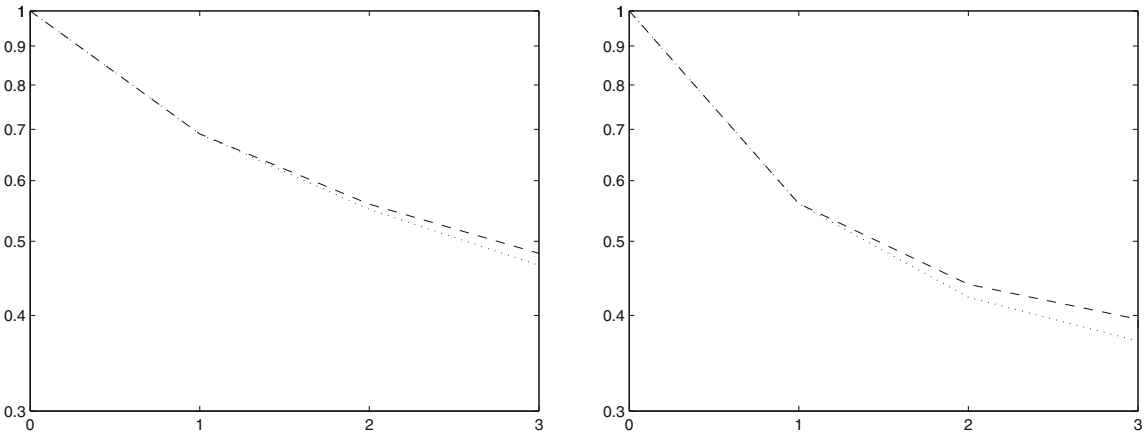


Fig. 1 GMRES relative residual norm curves for a one-dimensional convection-diffusion model problem with near Jordan block (*dashed lines*) and with true Jordan block (*dotted lines*). In the *left part* the right-hand side is e_n, in the *right part* it is $e_1 + e_n$

give explicit formulaes for the residual norms generated with this Jordan block using Theorem 3. Note that in this example $Y = C = I^-$.

– For $k = 1$, with Lemma 2, the numerator in (15) is

$$\sum_{I_2} |\det(C_{I_2,1:2})|^2.$$

There are six terms for I_2 : $\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}$, with only the last one giving the nonzero determinant $\det(C_{\{3,4\},\{1,2\}}) = -1$. For the denominator in (15) we sum over the trivial index sets $\{1\}, \{2\}, \{3\}, \{4\}$ and $\mathcal{I}_1 = \{1\}, \{2\}$. We obtain nonzero values for the index sets $\{3\},\{4\}$ only:

$$|\lambda \det(C_{\{3\},\{1\}}) + \det(C_{\{3\},\{2\}})|^2 = 1,\ |\lambda \det(C_{\{4\},\{1\}}) + \det(C_{\{4\},\{2\}})|^2 = |\lambda|^2.$$

The first residual norm satisfies

$$\|r_1\|^2 = \frac{1}{1 + |\lambda|^2}.$$

– For $k = 2$ the numerator in (15) is computed by summation over the sets of ordered indices $\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}$ with only the last one giving the nonzero determinant $\det(C_{\{2,3,4\},1:3}) = -1$.

For the denominator, we have $\mathcal{I}_2 = \{1, 2\}, \{1, 3\}, \{2, 3\}$, and the outer summation is over the index sets $\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}$. From these, only those not containing the index 1 lead to non-zero summands (the first three entries of the first row are all zero). Thus

$$\begin{aligned}
&\textstyle\sum_{I_2} \left|\lambda^2 \det(C_{I_2,\{1,2\}}) + \lambda \det(C_{I_2,\{1,3\}}) + \det(C_{I_2,\{2,3\}})\right|^2 \\
&= \left|\lambda^2 \det(C_{\{2,3\},\{1,2\}}) + \lambda \det(C_{\{2,3\},\{1,3\}}) + \det(C_{\{2,3\},\{2,3\}})\right|^2 \\
&+ \left|\lambda^2 \det(C_{\{2,4\},\{1,2\}}) + \lambda \det(C_{\{2,4\},\{1,3\}}) + \det(C_{\{2,4\},\{2,3\}})\right|^2 \\
&+ \left|\lambda^2 \det(C_{\{3,4\},\{1,2\}}) + \lambda \det(C_{\{3,4\},\{1,3\}}) + \det(C_{\{3,4\},\{2,3\}})\right|^2 \\
&= 1 + |\lambda|^2 + |\lambda|^4.
\end{aligned}$$

The square of the norm of the residual at iteration 2 is

$$\|r_2\|^2 = \frac{1}{1 + |\lambda|^2 + |\lambda|^4}.$$

– For $k = 3$ we have only one set of indices for I_4 that is, $\{1, 2, 3, 4\}$. Therefore,

$$\sum_{I_4} |\det(C_{I_4,1:4})|^2 = |\det(C)|^2 = |\det(I^-)|^2 = 1.$$

For the denominator in (15) we have $\mathcal{I}_3 = \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\} = I_3$. It yields

$$\begin{aligned}
&\sum_{I_3} |\lambda^3 \det(C_{I_3,\{1,2,3\}}) + \lambda^2 \det(C_{I_3,\{1,2,4\}}) + \lambda \det(C_{I_3,\{1,3,4\}}) \\
&+ \det(C_{I_3,\{2,3,4\}})|^2 = |\det(C_{\{1,2,3\},\{2,3,4\}})|^2 + |\lambda \det(C_{\{1,2,4\},\{1,3,4\}})|^2 \\
&+ |\lambda^2 \det(C_{\{1,3,4\},\{1,2,4\}})|^2 + |\lambda^3 \det(C_{\{2,3,4\},\{1,2,3\}})|^2 \\
&= 1 + |\lambda|^2 + |\lambda|^4 + |\lambda|^6
\end{aligned}$$

and the last non-zero residual norm satisfies

$$\|r_3\|^2 = \frac{1}{1 + |\lambda|^2 + |\lambda|^4 + |\lambda|^6}.$$

We can easily obtain formulaes for a right-hand side with more nonzero entries. For instance with a source term having the value $(\nu + h)/(-h^2)$ also in the last inner node $n/(n + 1)$, we obtain a linear system with a near Jordan block and right-hand side $e_1 + e_n$. The convergence curves for GMRES applied to the resulting system and applied to the same system where the nonzero lower subdiagonal entries have been replaced by zeros, are displayed in the right part of Fig. 1. They are very close. Note that the graphs represent relative residual norms or, equivalently, absolute residual norms for the systems where the right-hand side $e_1 + e_n$ was normalized through division with $\sqrt{2}$. Using Theorem 3 we obtain the exact residual norms for the system where the nonzero lower subdiagonal entries have been replaced by zeros (in this case $Y = C$ is the matrix $(I^- + e_1 e_1^T)$.)

- For $k = 1$, in comparison with the case $b = e_n$, the numerator in (15) contains the additional nonzero determinant $\det(C_{\{1,3\},\{1,2\}}) = b_1 = 1$. For the denominator in (15) we have an additonal nonzero value for the index sets $\{1\}$: $|\lambda \det(C_{\{1\},\{1\}}) + \det(C_{\{1\},\{2\}})|^2 = |\lambda b_1|^2 = |\lambda|^2$. The squared first relative residual norm is
$$\|r_1\|^2 = \frac{1}{1 + 2|\lambda|^2}.$$
- For $k = 2$, in comparison with the case $b = e_n$, the numerator in (15) also contains the nonzero determinant $\det(C_{\{1,2,3\},1:3}) = -b_1$. For the denominator, the outer summation is over the index sets $\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}$, $\{2, 4\}, \{3, 4\}$ where $\{1, 2\}, \{1, 3\}$ lead to the additional non-zero summands $|\lambda b_1|^2$ and $|\lambda^2 b_1|^2$, respectively. The square of the relative residual norm at iteration 2 is
$$\|r_2\|^2 = \frac{1}{1 + 2|\lambda|^2 + 2|\lambda|^4}.$$
- For $k = 3$, the numerator in (15) is $\sum_{I_4} |\det(C_{I_4,1:4})|^2 = |\det(C)|^2 = |\det(I^- + e_1 e_1^T)|^2 = 1$. For the denominator, the outer summand for the index set $\{1, 2, 3\}$ takes the value $|\lambda^3 b_1 + 1|^2$ and the remaining summands are unchanged. The last non-zero relative residual norm satisfies
$$\|r_3\|^2 = \frac{1}{2(|\lambda^3 + 1|^2 + |\lambda|^2 + |\lambda|^4 + |\lambda|^6)}.$$

We see that for these right-hand sides we would have good convergence if the modulus of λ is large, as one would expect. In other cases, however, it is in general not true that an eigenvalue close to zero hampers convergence for matrices with one Jordan block. If $\lambda \to 0$, then for a given k both the numerator and denominator in (15) go to values independent from λ. The speed of convergence is then fully determined by the entries of X and $X^{-1}b$ and need not be slow. In case it is not slow, our formulae give an explicit explanation for the limited role of the eigenvalue, i.e. of the theory in the series of papers [1, 17, 18].

4 GMRES for non-diagonalizable matrices

The generalization of Section 3 to multiple Jordan blocks is straightforward. Let A have the Jordan form XJX^{-1} and let it have m ($m \leq n$) distinct eigenvalues denoted as $\lambda_1, \lambda_2, \ldots, \lambda_m$. We assume A is non-derogatory because we consider GMRES processes that do not terminate before iteration n. Let the size of the Jordan block J_i corresponding to λ_i be n_i, i.e. $\sum_{i=1}^{m} n_i = n$, and let us denote by s_i, $i = 1, \ldots, m$ the index of the row where the block J_i starts, to which we add $s_{m+1} = n + 1$. The block J_i goes from row s_i to row $s_{i+1} - 1$. To avoid early termination, we also assume that the right-hand side b is a vector of unit norm such that the entries on positions $s_{i+1} - 1$, $1 \leq i \leq m$, of $c = X^{-1}b$ are nonzero.

As before, we have

$$M = K^*K = \left(c \;\; Jc \;\cdots\; J^{n-1}c\right)^* X^*X \left(c \;\; Jc \;\cdots\; J^{n-1}c\right).$$

For multiple Jordan blocks, the decomposition (13) can be modified as follows. If we define the rows s_i to $s_{i+1} - 1$ of E corresponding to the eigenvalue λ_i as

$$E_{s_i:s_{i+1}-1,:} \equiv \begin{pmatrix} 1 & \lambda_i & \lambda_i^2 & \cdots & \lambda_i^{n_i-1} & \cdots & \lambda_i^{n-1} \\ 0 & 1 & 2\lambda_i & \cdots & \binom{n_i-1}{1}\lambda_i^{n_i-2} & \cdots & \binom{n-1}{1}\lambda_i^{n-2} \\ 0 & 0 & 1 & \cdots & \binom{n_i-1}{2}\lambda_i^{n_i-3} & \cdots & \binom{n-2}{2}\lambda_i^{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & \cdots & \binom{n-1}{n_i-1}\lambda_i^{n-n_i} \end{pmatrix}$$

and the corresponding diagonal block of C as

$$C_{s_i:s_{i+1}-1,s_i:s_{i+1}-1} \equiv \begin{pmatrix} c_{s_i} & c_{s_i+1} & \cdots & \cdots & c_{s_{i+1}-1} \\ c_{s_i+1} & c_{s_i+2} & \cdots & c_{s_{i+1}-1} & \\ c_{s_i+2} & \cdots & c_{s_{i+1}-1} & & \\ \vdots & c_{s_{i+1}-1} & & & \\ c_{s_{i+1}-1} & & & & \end{pmatrix},$$

then

$$\left(c \;\; Jc \;\cdots\; J^{n-1}c\right) = CE.$$

The matrix C is block diagonal with Hänkel anti-upper triangular diagonal blocks of order n_i. We again give an example to illustrate.

Consider a matrix $A = XJX^{-1}$ of order 5 with J defined as

$$J = \begin{pmatrix} \lambda & 1 & & & \\ & \lambda & 1 & & \\ & & \lambda & & \\ & & & \mu & 1 \\ & & & & \mu \end{pmatrix}, \tag{17}$$

where λ and μ ($\lambda \neq \mu$) are given complex numbers different from 0. Let $c = X^{-1}b$, where b is the right-hand side, and let c have no zero entries. Then the Krylov matrix $\begin{pmatrix} c & Jc & \cdots & J^{n-1}c \end{pmatrix}$ is

$$\begin{pmatrix} c_1 & \lambda c_1 + c_2 & \lambda^2 c_1 + 2\lambda c_2 + c_3 & \lambda^3 c_1 + 3\lambda^2 c_2 + 3\lambda c_3 & \lambda^4 c_1 + 4\lambda^3 c_2 + 6\lambda^2 c_3 \\ c_2 & \lambda c_2 + c_3 & \lambda^2 c_2 + 2\lambda c_3 & \lambda^3 c_2 + 3\lambda^2 c_3 & \lambda^4 c_2 + 4\lambda^3 c_3 \\ c_3 & \lambda c_3 & \lambda^2 c_3 & \lambda^3 c_3 & \lambda^4 c_3 \\ c_4 & \mu c_4 + c_5 & \mu^2 c_4 + 2\mu c_5 & \mu^3 c_4 + 3\mu^2 c_5 & \mu^4 c_4 + 4\mu^3 c_5 \\ c_5 & \mu c_5 & \mu^2 c_5 & \mu^3 c_5 & \mu^4 c_5 \end{pmatrix}$$

and can be factorized as

$$\begin{pmatrix} c & Jc & \cdots & J^{n-1}c \end{pmatrix} = \begin{pmatrix} c_1 & c_2 & c_3 & 0 & 0 \\ c_2 & c_3 & 0 & 0 & 0 \\ c_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_4 & c_5 \\ 0 & 0 & 0 & c_5 & 0 \end{pmatrix} \begin{pmatrix} 1 & \lambda & \lambda^2 & \lambda^3 & \lambda^4 \\ 0 & 1 & 2\lambda & 3\lambda^2 & 4\lambda^3 \\ 0 & 0 & 1 & 3\lambda & 6\lambda^2 \\ 1 & \mu & \mu^2 & \mu^3 & \mu^4 \\ 0 & 1 & 2\mu & 3\mu^2 & 4\mu^3 \end{pmatrix},$$

with a block diagonal matrix C.

Let, as before,

$$Y \equiv XC.$$

Then if the $(k+1)$st leading principal submatrix M_{k+1} of M is written as

$$\begin{aligned} M_{k+1} &= \begin{pmatrix} c & Jc & \cdots & J^k c \end{pmatrix}^* X^* X \begin{pmatrix} c & Jc & \cdots & J^k c \end{pmatrix} \\ &= E^*_{:,1:k+1} C^* X^* X C E_{:,1:k+1} = (Y E_{:,1:k+1})^* Y E_{:,1:k+1}, \end{aligned}$$

we immediately obtain, again by using the proof technique of Theorem 1, the formula

$$\|r_k\|^2 = \frac{\sum_{I_{k+1}} |\sum_{J_{k+1}} \det(Y_{I_{k+1},J_{k+1}}) \det(E_{J_{k+1},1:k+1})|^2}{\sum_{I_k} \left|\sum_{J_k} \det(Y_{I_k,J_k}) \det(E_{J_k,2:k+1})\right|^2}. \tag{18}$$

The formula is the same as the one presented in Corollary 1, but of course, Y and E are here generalizations of the Y and E in Corollary 1. E represents all the influence of eigenvalues and Y all the influence of eigenvectors, principal vectors and right-hand side. The remarks in Sections 2 and 4 on the role of $\kappa(X)$ and of $X^{-1}b$ apply to this section, too.

A difference is that the interplay between the distinct eigenvalues will play a role. The determinants of $E_{J_{k+1},1:k+1}$ and $E_{J_k,2:k+1}$ may contain eigenvalue differences. For example, so do most determinants of E involved in forming $\|r_3\|^2$ for the matrix J in (17), see Tables 1 and 2. All determinants in Table 1 have $\mu - \lambda$ as a factor. Hence

Table 1 Determinants of $E_{J_4,1:4}$ for the numerator in (18) with $k = 3$, for the matrix J in (17)

Indices in J_4	value
{1,2,3,4}	$(\mu - \lambda)^3$
{1,2,3,5}	$3(\mu - \lambda)^2$
{1,2,4,5}	$(\mu - \lambda)^4$
{1,3,4,5}	$-2(\mu - \lambda)^3$
{2,3,4,5}	$3(\mu - \lambda)^2$

they may be small if μ is close to λ. This suggests that eigenvalue clusters accelerate convergence whereas outliers cause delay, which is often true (see, e.g., [4]). If $\mu = \lambda$, corresponding to two Jordan blocks with the same eigenvalue, we have early termination, $\|r_3\| = 0$ (in exact arithmetic).

We now investigate whether with non-diagonalizable matrices, GMRES residual norms are slightly less dependent on eigenvalues than with diagonalizable matrices in the sense that not all summands in (18) depend upon eigenvalues. We have seen with Theorem 3 that this holds for matrices with a single Jordan block.

For simplicity, we first we address the case $k = 1$. Let us consider the determinants in the numerator of (18), i.e. the determinants of $E_{J_2,\{1,2\}}$ for the set of indices J_2. There are $n!/(2(n-2)!)$ of them. But the rows that are involved are only of three different types whatever the dimension n is. The first type that we can denote as $t_1(\lambda_i)$ is $t_1(\lambda_i) = \begin{pmatrix} 1 & \lambda_i \end{pmatrix}$, for an eigenvalue λ_i. The two other types are $t_2 = \begin{pmatrix} 0 & 1 \end{pmatrix}$ and $t_3 = \begin{pmatrix} 0 & 0 \end{pmatrix}$. The two last types may or may not exist depending on the values of n_i, $i = 1, \ldots, m$. We have only three kinds of non-zero determinants

$$\begin{vmatrix} 1 & \lambda_i \\ 1 & \lambda_j \end{vmatrix} = \lambda_j - \lambda_i, \quad \begin{vmatrix} 1 & \lambda_i \\ 0 & 1 \end{vmatrix} = 1, \quad \begin{vmatrix} 0 & 1 \\ 1 & \lambda_i \end{vmatrix} = -1. \tag{19}$$

Then in the terms

$$\left| \sum_{J_2} \det(Y_{I_2,J_2}) \det(E_{J_2,1:2}) \right|^2,$$

of the numerator of (18), the sum runs over the set of indices such that $\det(E_{J_2,1:2}) \neq 0$ that is, such that we have one of the three kinds of determinant listed above. With the

Table 2 Determinants of $E_{J_3,2:4}$ for the denominator in (18) with $k = 3$, for the matrix J in (17)

Indices in J_3	value	Indices in J_3	value
{1,2,3}	λ^3	{1,4,5}	$\lambda\mu^2(\mu - \lambda)^2$
{1,2,4}	$\lambda^2\mu(\mu - \lambda)^2$	{2,3,4}	$\mu[(\mu - \lambda)^2 + \lambda(2\lambda - \mu)]$
{1,2,5}	$\lambda^2(\mu - \lambda)(3\mu - \lambda)$	{2,3,5}	$3(\mu - \lambda)^2$
{1,3,4}	$\lambda\mu(\mu - \lambda)(\mu - 2\lambda)$	{2,4,5}	$\mu^2(\mu - \lambda)(\mu - 3\lambda)$
{1,3,5}	$\lambda[2(\mu - \lambda)^2 + \mu(\mu - 2\lambda)]$	{3,4,5}	$\mu^2(3\lambda - 2\mu)$

second and third kind there is no dependence on eigenvalues. For the denominator of (18) we can proceed similarly. Thus, depending on the sizes of the individual Jordan blocks, a number of summands is independent from the spectrum.

For $k > 1$ we have the following straightforward result.

Proposition 3 *If $k < \max_i(n_i)$, then in formula (18) there are determinants of both $E_{J_{k+1},1:k+1}$ and $E_{J_k,2:k+1}$ that are equal to 1.*

Proof The result is obvious since some of the submatrices are upper triangular with ones on the diagonal. □

It is not difficult to see that when an eigenvalue approaches zero, this gives determinants tending to be independent on that eigenvalue. Similarly to the previous section, the influence of the corresponding Jordan block on GMRES is then fully determined by the right-hand side and eigenvectors and/or principal vectors and consequently, eigenvalues close to the origin do not seem to necessarily hamper convergence.

5 Conclusion

We presented the solution of the minimization problem (1) for GMRES residual norms generated with general diagonalizable and with non-diagonalizable matrices. It is explicitly formulated in a closed form, unlike the norms of the GMRES residuals in GMRES computations. The solution is not simple and has no immediate practical application but it completely describes the mechanism of forming the residual norm from eigenvalues, eigenvectors or principal vectors and the right-hand side. It shows in what (complicated) way eigenvalues influence GMRES convergence. Other objects than eigenvalues may lead to more elegant formulaes, but if we wish to know the exact influence of eigenvalues, the presented closed-form expressions give the answer. In the diagonalizable case, it is eigenvalue products and products of eigenvalue differences that influence the residual norm. In the non-diagonalizable case, more general polynomials in eigenvalues play a role in forming the residual norm and small eigenvalues are less prone to hamper convergence. Eigenvectors (principal vectors) influence residual norms in two ways. Determinants of the eigenvector (principal vector) matrix play the most important role. The inverse of this matrix contributes only in the form of its product with the right-hand side. As for the right-hand side, it contributes only through its components in the eigenvector (principal vector) basis. The degree to which GMRES is governed by eigenvalues is not so much determined by the departure from diagonalizability of the system matrix, but in general more by the departure from orthogonality of the eigenvector (principal vector) matrix X. With a small value of $\kappa(X)$, GMRES is governed by the spectrum even if the system matrix is defective; with a larger value of $\kappa(X)$ GMRES may or may not be governed by the spectrum, depending on X, $X^{-1}b$ and the interplay between them.

Future work includes extension to other Krylov methods.

Acknowledgments We thank Zdeněk Strakoš for stimulating work in this direction and an anonymous referee for helping to improve the presentation of this paper. The work of the second author was supported by the institutional support RVO:67985807 and by the grant GA13-06684S of the Grant Agency of the Czech Republic.

References

1. Arioli, M., Pták, V., Strakoš, Z.: Krylov sequences of maximal length and convergence of GMRES. BIT **38**(4), 636–643 (1998)
2. Baglama, J., Calvetti, D., Golub, G.H., Reichel, L.: Adaptively preconditioned GMRES algorithms. SIAM J. Sci. Comput. **20**(1), 243–269 (1998)
3. Bellalij, M., Jbilou, K., Sadok, H.: New convergence results on the global GMRES method for diagonalizable matrices. J. Comput. Appl. Math. **219**, 350–358 (2008)
4. Campbell, S.L., Ipsen, I.C.F., Kelley, C.T., Meyer, C.D.: GMRES and the minimal polynomial. BIT **36**(4), 664–675 (1996)
5. Carpentieri, B., Duff, I.S., Giraud, L.: A class of spectral two-level preconditioners. SIAM J. Sci. Comput. **25**(2), 749–765 (2003)
6. Carpentieri, B., Giraud, L., Gratton, S.: Additive and multiplicative two-level spectral preconditioning for general linear systems. SIAM J. Sci. Comput. **29**(4), 1593–1612 (2007)
7. Chapman, A., Saad, Y.: Deflated and augmented Krylov subspace techniques. Numer. Linear Algebra Appl. **4**(1), 43–66 (1997)
8. Duintjer Tebbens, J., Meurant, G.: Any Ritz value behavior is possible for Arnoldi and for GMRES. SIAM J. Matrix Anal. Appl. **33**(3), 958–978 (2012)
9. Duintjer Tebbens, J., Meurant, G.: Prescribing the behavior of early terminating GMRES and Arnoldi iterations. Num. Alg. **65**(1), 69–90 (2014)
10. Duintjer Tebbens, J., Meurant, G., Sadok, H., Strakoš, Z.: On investigating GMRES convergence using unitary matrices. Lin. Alg. Appl. **450**, 83–107 (2014)
11. Eiermann, M.: Fields of values and iterative methods. Lin. Alg. Appl. **180**, 167–197 (1993)
12. Erhel, J., Burrage, K., Pohl, B.: Restarted GMRES preconditioned by deflation. J. Comput. Appl. Math. **69**(2), 303–318 (1996)
13. Gautschi, W.: On inverses of Vandermonde and confluent Vandermonde matrices. III. Numer. Math. **29**, 445–450 (1977/78)
14. Giraud, L., Gratton, S., Martin, E.: Incremental spectral preconditioners for sequences of linear systems. Appl. Numer. Math. **57**(11–12), 1164–1180 (2007)
15. Giraud, L., Gratton, S., Pinel, X., Vasseur, X.: Flexible GMRES with deflated restarting. SIAM J. Sci. Comput. **32**(4), 1858–1878 (2010)
16. Greenbaum, A.: Generalizations of the field of values useful in the study of polynomial functions of a matrix. Lin. Alg. Appl. **347**, 233–249 (2002)
17. Greenbaum, A., Pták, V., Strakoš, Z.: Any nonincreasing convergence curve is possible for GMRES. SIAM J. Matrix Anal. Appl. **17**(3), 465–469 (1996)
18. Greenbaum, A., Strakoš, Z.: Matrices that generate the same Krylov residual spaces. In: Recent Advances in Iterative Methods, volume 60 of IMA Vol. Math. Appl., pp. 95–118. Springer, New York (1994)
19. Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. J. Research Nat. Bur. Standards **49**, 409–436 (1952)
20. Huhtanen, M., Nevanlinna, O.: Minimal decompositions and iterative methods. Numer. Math. **86**(2), 257–281 (2000)
21. Ipsen, I.C.F.: Expressions and bounds for the GMRES residual. BIT **40**(3), 524–535 (2000)
22. Kharchenko, S.A., Yu. Yeremin, A.: Eigenvalue translation based preconditioners for the GMRES(k) method. Numer. Linear Algebra Appl. **2**(1), 51–77 (1995)
23. Kuijlaars, A.B.J.: Convergence analysis of Krylov subspace iterations with methods from potential theory. SIAM Rev. **48**(1), 3–40 (2006)
24. Le Calvez, C., Molina, B.: Implicitly restarted and deflated GMRES. Numer. Algorithms **21**(1–4), 261–285 (1999). Numerical methods for partial differential equations (Marrakech, 1998)
25. Liesen, J., Rozložník, M., Strakoš, Z.: Least squares residuals and minimal residual methods. SIAM J. Sci. Comput. **23**(5), 1503–1525 (2002)

26. Liesen, J., Strakoš, Z.: Convergence of GMRES for tridiagonal Toeplitz matrices. SIAM J. Matrix Anal. Appl. **26**(1), 233–251 (2004)
27. Liesen, J., Tichý, P.: Convergence analysis of Krylov subspace methods. GAMM Mitt. Ges. Angew Math. Mech. **27**(2), 153–173 (2004)
28. Liesen, J., Tichý, P.: The worst-case GMRES for normal matrices. BIT **44**(1), 79–98 (2004)
29. Loghin, D., Ruiz, D., Touhami, A.: Adaptive preconditioners for nonlinear systems of equations. J. Comput. Appl. Math. **189**(1–2), 362–374 (2006)
30. Morgan, R.B.: A restarted GMRES method augmented with eigenvectors. SIAM J. Matrix Anal. Appl. **16**(4), 1154–1171 (1995)
31. Morgan, R.B.: Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations. SIAM J. Matrix Anal. Appl. **21**(4), 1112–1135 (2000)
32. Morgan, R.B.: GMRES with deflated restarting. SIAM J. Sci. Comput. **24**(1), 20–37 (2002)
33. Nachtigal, N.M., Reddy, S.C., Trefethen, L.N.: How fast are nonsymmetric matrix iterations? SIAM J. Matrix Anal. Appl. **13**(3), 778–795 (1992)
34. Paige, C.C., Saunders, M.A.: LSQR: An algorithm for sparse linear equations and sparse least squares. ACM Trans. Math. Softw. **8**(1), 43–71 (1982)
35. Parks, M.L., de Sturler, E., Mackey, G., Johnson, D.D., Maiti, S.: Recycling Krylov subspaces for sequences of linear systems. SIAM J. Sci. Comput. **28**(5), 1651–1674 (2006)
36. Pestana, J., Wathen, A.: On choice of preconditioner for minimum residual methods for non-hermitian matrices. J. Comput. Appl. Math. **249**, 57–68 (2013)
37. Saad, Y.: Iterative methods for sparse linear systems. Society for Industrial and Applied Mathematics, 2nd edn. Philadelphia (2003)
38. Saad, Y., Schultz, M.H.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **7**(3), 856–869 (1986)
39. Sadok, H.: Analysis of the convergence of the minimal and the orthogonal residual methods. Numer. Algorithms **40**(2), 201–216 (2005)
40. Stewart, G.W.: Collinearity and least squares regression. Stat. Sci. **2**(1), 68–100 (1987)
41. Stynes, M.: Steady-state convection-diffusion problems. Acta Numer. **14**, 445–508 (2005)
42. Tichý, P., Liesen, J., Faber, V.: On worst-case GMRES, ideal GMRES, and the polynomial numerical hull of a Jordan block. Electron. Trans. Numer. Anal. **26**, 453–473 (2007)
43. Titley-Peloquin, D., Pestana, J., Wathen, A.: GMRES convergence bounds that depend on the right-hand side vector. IMA J. Numer. Anal. **34**(2), 462–479 (2014)
44. Trefethen, L.N., Embree, M.: Spectra and Pseudospectra. Princeton University Press, Princeton (2005)
45. Zítko, J.: Generalization of convergence conditions for a restarted GMRES. Numer. Linear Algebra Appl. **7**, 117–131 (2000)

SIAM J. Matrix Anal. Appl.
Vol. 35, No. 1, pp. 174–197

ON INCREMENTAL CONDITION ESTIMATORS IN THE 2-NORM*

JURJEN DUINTJER TEBBENS† AND MIROSLAV TŮMA‡

Abstract. This paper deals with estimating the condition number of triangular matrices in the Euclidean norm. The two main incremental methods, based on the work of Bischof and the later work of Duff and Vömel, are compared. The paper presents new theoretical results revealing their similarities and differences. As typical in condition number estimation, there is no universal always-winning strategy, but theoretical and experimental arguments show that the clearly preferable approach is the algorithm of Duff and Vömel when appropriately applied to both the triangular matrix itself and its inverse. This leads to a highly accurate incremental condition number estimator.

Key words. condition number estimation, matrix inverses, incremental condition estimator, incremental norm estimator

AMS subject classifications. 15A18, 65F35

DOI. 10.1137/130922872

1. Introduction.

The condition number

$$\kappa(A) = \|A\| \cdot \|A^{-1}\|$$

of a nonsingular matrix is a very important quantity in numerical linear algebra. While its computation is typically as expensive as solving a corresponding system of linear equations, there exist efficient procedures for condition number estimation. Proper use of the computed estimates can often save a lot of computational effort.

First, matrix condition number estimates may be used in the basic tasks of numerical linear algebra, that is, in solving systems of linear algebraic equations and solving eigenvalue problems, to assess the quality of the computed solutions and their sensitivity to perturbations. Further, there are specific fields in scientific computing that are strongly linked with condition number estimation. The estimated condition number may be used to monitor and control adaptive computational processes, sometimes using the terminology adaptive condition estimators (ACE). Such adaptive processes may include evaluation of adaptive filters [23], [30] and recursive least squares in signal processing [21] or solving nonlinear problems by linearization methods [23], [35]. Standard algebraic approaches are used for tracking the condition number in a sequence of modified matrices of the same dimension as well as when matrices are subsequently constructed by augmentation [33], [34], [36], [20], [21]. ACE based on properties of model and grid hierarchies is a standard tool in multilevel PDE solvers [28]. Another type of problem-oriented ACE in recursive least squares measured with a norm close to the Frobenius norm is represented in [1], [2]. An emerging application is the use of condition number estimates for dropping and pivoting in incomplete matrix decompositions, which we will mention later.

*Received by the editors May 29, 2013; accepted for publication (in revised form) by J. L. Barlow November 5, 2013; published electronically February 12, 2014. This work was supported by RVO:67985807 and by grant GA13-06684S of the Grant Agency of the Czech Republic.
http://www.siam.org/journals/simax/35-1/92287.html

†Institute of Computer Science, Academy of Sciences of the Czech Republic, 18207 Praha 8, Czech Republic, and Faculty of Pharmacy in Hradec Králové, Charles University in Prague, 500 05 Hradec Králové, Czech Republic (duintjertebbens@cs.cas.cz).

‡Institute of Computer Science, Academy of Sciences of the Czech Republic, 18207 Praha 8, Czech Republic (tuma@cs.cas.cz).

In order to have useful condition estimators, they should be cheap. At the same time, they should provide condition number approximations which are reasonably accurate, and this may mean different things in different applications. Sometimes, relatively rough estimates are satisfactory, e.g., it is sufficient in many cases that the estimates stay within a reasonable multiplicative factor from the exact condition number; see, e.g., [16]. In other cases, more precise estimates may be needed [27].

Condition number estimators typically provide lower bounds on the condition number of a nonsingular matrix A by estimating a lower bound on the norm of A and an upper bound on the norm of A^{-1}. The most popular general approaches compute approximations of the condition number in the 1-norm [22], [25], [24], [26]. An important milestone in the development of estimators in the 2-norm was the incremental condition estimation (ICE) of a triangular matrix that was introduced in papers by Bischof [3] and Bischof, Lewis, and Pierce [5] and further generalized for solving related tasks [6], [36]. This strategy is naturally connected to adaptive techniques and contains clearly visible links to matrix decompositions. As mentioned by Stewart [37], the approach can be viewed as a special case of the framework in [14]. A closely related approach called incremental norm estimation (INE) was developed by Duff and Vömel in [18] to get an estimation of the norm of a triangular matrix. While a slight reformulation of this algorithm similar to that in [3] can be used to estimate the minimum singular value as well, we will see in this paper that this does not work well in practice and we will give a partial explanation for this. Nevertheless, when the inverses of the triangular factors of A are available, INE could be used to get a useful estimate for the minimum singular value of A [18]. A similar conclusion follows for the recent iterative procedure to get a lower bound for the minimum singular value given in the interesting paper [29]. The actual strategy is based on an improvement of the algorithm in [19] and applied to symmetric and positive definite Toeplitz matrices. ICE is also closely related to rank-revealing decompositions; see, e.g., [32].

A strong motivation to study and further develop incremental condition estimators is their applicability in incomplete decompositions. In particular, a part of recent advances in preconditioning of systems of linear algebraic equations is based on monitoring the conditioning of the partially computed factors via a condition estimator. The incremental nature of the estimator enables one to monitor and control both dropping and pivoting of the decomposition. This is done in strategies developed by Bollhöfer [7], [8] and Bollhöfer and Saad [10] and implemented in the package ILUPACK [9]; see also their use in the multilevel framework [11]. Both perturbation arguments and experiments point out that preconditioners from incomplete decompositions using dropping criteria based on conditioning control are rather robust, but we believe that more accurate incremental strategies may help to push the approach even further. Note that the use of ICE for pivoting in decompositions was considered much earlier (see, e.g., [4]), but the significant progress in this research direction is connected with the work of Bollhöfer and Saad.

Recently, incomplete decompositions that compute both direct and inverse factors were introduced. That is, they compute not only the standard Cholesky or LU factors but also their inverses. In [38] the author proposes to compute the inverse of the incomplete factor once the direct factor is computed. The mixed direct-inverse decompositions in [12], [13] obtain the direct and inverse factors simultaneously, enabling one to exploit information from the partial inverse factor for the computation of the direct factor and vice versa. It was shown that despite rather sophisticated implementation, typical computational costs of the decomposition may still be low.

Moreover, condition estimators can be applied to both the direct and the inverse factor, thus enabling one to use the more accurate condition estimators discussed in this paper.

This paper presents some new theoretical and practical results leading to an improved incremental condition estimator in the 2-norm. As it is well known that the strengths of different condition estimators are often complementary and any one of them can sometimes fail, we do not propose a strategy that is always better than all the other approaches, but we have rather strong theoretical and computational evidence to propose a choice based on INE. In the paper, we will show some theoretical results related to the condition estimators introduced in [3] and [18] as well as the mutual relation of these estimators. In particular, we will show that the best strategy should be based on the computed factor as well as its inverse. We recall that factorizations that could be used for this task are readily available. The paper is organized as follows. In section 2 the two basic strategies for ICE in the 2-norm are introduced. Section 3 provides new theoretical results on the ICE and INE estimators. In particular, it reveals the strong potential of the INE algorithm using the factor as well as its inverse. Section 4 then analyzes reasons for the superiority of INE over ICE that is clear from both the graphical demonstration in this section and from the numerical experiments in section 5. In the sections to follow, we will assume that A is real and $\|\cdot\|$ will denote the 2-norm. With "direct factor" we will mean a triangular Cholesky, L, or U factor of a given input matrix, as opposed to its inverse, the "inverse factor."

2. Incremental condition estimators in the 2-norm. This section presents a brief overview of the two basic incremental strategies to estimate the 2-norm condition number of a triangular matrix. The idea is to find an upper bound estimate σ_{min}^{est} of its smallest singular value and a lower bound estimate σ_{max}^{est} of its largest singular value. The condition number estimate is then $\sigma_{max}^{est}/\sigma_{min}^{est}$. Without loss of generality we assume our matrix to be upper triangular. By the *incremental* nature of the estimates we mean that the estimate for the leading principal submatrix $\hat{R}$ of dimension $k+1$ is obtained from the estimate for its leading principal submatrix R of dimension k in a simple way, without explicitly accessing the entries of R. In order to be able to do this, we also keep estimates of the corresponding singular vectors. Note that the basic matrix decompositions like Cholesky or LU reveal the triangular factors just in this incremental way and the incremental estimates may be used not only to form the final condition number estimate, but they may be exploited throughout the decomposition.

Let us use the following notation:

$$\hat{R} = \begin{bmatrix} R & v \\ 0 & \gamma \end{bmatrix}. \tag{2.1}$$

As mentioned above, the first incremental estimation strategy of this kind, ICE, was proposed by Bischof [3] in 1990. This method computes approximations to the extremal singular values and to left singular vectors of triangular leading principal submatrices. Note that if $R = U\Sigma V^T$ is the SVD of R, an extremal left singular vector u_{ext} satisfies $\|u_{ext}^T R\| = \|u_{ext}^T U\Sigma V^T\| = \sigma_{ext}(R)$ with σ_{ext} denoting the extremal singular value. The ICE method computes

$$\sigma_{ext}^C(R) = \|y_{ext}^T R\| \approx \sigma_{ext}(R),$$

where ext is substituted for either min or max and y_{ext} denotes a left singular vector approximation. The superscript C here means the considered ICE incremental strategy that can be described as follows. Consider the submatrix $\hat{R}$. The algorithm computes the approximation $\sigma_{ext}^C(\hat{R})$ from the optimization problem

$$\|\hat{y}_{ext}^T\hat{R}\| = \operatorname{ext}_{\|[s,c]\|=1} \left\| \begin{bmatrix} s\, y_{ext}^T, & c \end{bmatrix} \begin{bmatrix} R & v \\ 0 & \gamma \end{bmatrix} \right\| = \left\| \begin{bmatrix} s_{ext}\, y_{ext}^T, & c_{ext} \end{bmatrix} \begin{bmatrix} R & v \\ 0 & \gamma \end{bmatrix} \right\|,$$

where the approximation $\hat{y}_{ext}$ of the left singular vector of $\hat{R}$ is

$$\hat{y}_{ext} \equiv \begin{bmatrix} s_{ext}\, y_{ext} \\ c_{ext} \end{bmatrix}.$$

It can be easily verified that s_{ext} and c_{ext} are the components of the eigenvector corresponding to the extremal (minimum or maximum) eigenvalue of the matrix

$$B_{ext}^C \equiv \begin{bmatrix} \sigma_{ext}^C(R)^2 + (y_{ext}^T v)^2 & \gamma(y_{ext}^T v) \\ \gamma(y_{ext}^T v) & \gamma^2 \end{bmatrix}. \tag{2.2}$$

If B_{ext}^C has two identical eigenvalues, the algorithm of [3] puts $s_{ext} = 0$ and $c_{ext} = 1$. Further,

$$\sigma_{ext}^C(\hat{R}) \equiv \|\hat{y}_{ext}^T\hat{R}\| = \sqrt{\lambda_{ext}(B_{ext}^C)},$$

where λ_{ext} denotes the extremal (minimum or maximum) eigenvalue. Clearly, the involved eigenvectors are computed without accessing R. Note that the original derivation in [3] uses a lower triangular matrix and it is slightly different from the one presented here; see [18].

Another incremental strategy was proposed in 2002 by Duff and Vömel [18] and used only for norm estimation based on a maximization problem, although it is possible to formulate the dual minimization problem as well; it is denoted here by INE using the superscript N. It computes approximations $\sigma_{ext}^N(R)$ of the extremal singular values

$$\sigma_{ext}^N(R) = \|Rz_{ext}\| \approx \sigma_{ext}(R)$$

as well as the corresponding INE approximations z_{ext} to the *right* singular vectors. Similarly as above, $\sigma_{ext}^N(\hat{R})$ is obtained from the optimization problem

$$\|\hat{R}\hat{z}_{ext}\| = \operatorname{ext}_{\|[s,c]\|=1} \left\| \begin{bmatrix} R & v \\ 0 & \gamma \end{bmatrix} \begin{bmatrix} s\, z_{ext} \\ c \end{bmatrix} \right\| = \left\| \begin{bmatrix} R & v \\ 0 & \gamma \end{bmatrix} \begin{bmatrix} s_{ext} z_{ext} \\ c_{ext} \end{bmatrix} \right\|,$$

where the approximation $\hat{z}_{ext}$ of the right singular vector of $\hat{R}$ is

$$\hat{z}_{ext} \equiv \begin{bmatrix} s_{ext}\, z_{ext} \\ c_{ext} \end{bmatrix}.$$

The scalars s_{ext} and c_{ext} are then the entries of the eigenvector corresponding to the extremal (minimum or maximum) eigenvalue of the matrix

$$B_{ext}^N \equiv \begin{bmatrix} \sigma_{ext}^N(R)^2 & z_{ext}^T R^T v \\ z_{ext}^T R^T v & v^T v + \gamma^2 \end{bmatrix} \tag{2.3}$$

with the convention that $s_{ext} = 0$ and $c_{ext} = 1$ when B^N_{ext} has two identical eigenvalues. Then

$$\sigma^N_{ext}(\hat{R}) \equiv \|\hat{R}\hat{z}_{ext}\| = \sqrt{\lambda_{ext}(B^N_{ext})}.$$

In the remaining text we will further simplify the notation as follows. The subscripts min and max denoting minimum or maximum, respectively, such as s_{max} or y_{min}, will be replaced by plus or minus signs, which give in this example $s_+ \equiv s_{max}$ and $y_- \equiv y_{min}$.

Note that the main costs involved in both techniques come from the inner products needed to get the entries of the matrices B^C_{ext} and B^N_{ext}. For a dense triangular matrix of dimension n the total costs to obtain its estimate are of the order n^2. Further, the above descriptions give no clear indication about whether one technique is superior to the other. In [18] the authors conclude, based on their experiments, that there is no general superiority of one technique. They explain that INE is more suited for sparse matrices and they show experimentally that INE is slightly superior for finding the largest singular value of dense triangular matrices. The following sections contain, among others, new theoretical comparisons of the quality of the two described techniques and a strong numerical confirmation of our findings.

3. ICE and INE estimates using both direct and inverse factors. Let us consider ICE and INE in the situation when we have both the direct triangular factor and its inverse available. In this section we are interested to know whether exploiting the inverse factor may help to improve accuracy of the estimates. At first sight this may seem trivial since the hard part in the estimation is often to find a good approximation of the minimum singular value. If the inverse is available, the problem can be circumvented by estimating the maximum singular value of the inverse. However, we will see that the two considered techniques behave differently in this respect.

Note that the inverse or its approximation is naturally available in the mixed direct-inverse decompositions [12], [13] mentioned in the introduction. In addition, information on rows and/or columns of the inverse is computed when applying the techniques of [7], [8], [10]. In some other applications, for example, in signal processing [15], [31], it is necessary to compute the matrix inverses explicitly and this is traditionally done via their triangular factors. Further, the inversion of a triangular factor can be done at costs that are low compared to the computation of the factor. For example, the algorithm in [18, Lemma 3.1] asks for about $n^3/6$ flops; see also the techniques in [38].

First we will show that using the inverse triangular factor does not give any improvement for ICE. Let us start with a simple lemma related to the exact singular values and vectors.

LEMMA 3.1. *Let R be a nonsingular matrix. Then the extremal singular values of R and R^{-1} satisfy $\sigma_-(R) = 1/\sigma_+(R^{-1})$. The corresponding left singular vectors y_- and x_+ of R and R^{-1}, respectively, satisfy*

$$\sigma_-(R)x_+^T = y_-^T R. \tag{3.1}$$

Proof. The first part of the assertion is trivial. Let $R = USW^T$ be the SVD of R with the singular values in S in nonascending order. Then $R^{-1} = WS^{-1}U^T$ and the left singular vectors y_- and x_+ can be expressed as $y_- = Ue_n$ and $x_+ = We_n$, respectively. Then we can write $x_+^T R^{-1} = e_n^T W^T R^{-1} = e_n^T W^T W S^{-1} U^T = (1/\sigma_-(R))e_n^T U^T = (1/\sigma_-(R))y_-^T$, which implies (3.1). ∎

The main result relating the ICE estimates for R and R^{-1} looks similarly.

THEOREM 3.2. *Let R be a nonsingular upper triangular matrix. Then the ICE estimates of the singular values of R and R^{-1} satisfy*

$$\sigma_-^C(R) = 1/\sigma_+^C(R^{-1}). \tag{3.2}$$

The approximate left singular vectors y_- and x_+ corresponding to the ICE estimates for R and R^{-1}, respectively, satisfy

$$\sigma_-^C(R)x_+^T = y_-^T R. \tag{3.3}$$

Proof. Consider mathematical induction on the dimension n of R. Clearly, the estimates are exact for $n = 1, 2$. Assume that the lemma holds for some $n \geq 2$ and we will prove it for $n+1$. Let us use the notation (2.1) for the upper triangular $\hat{R}$ of dimension $n+1$. The estimate $\sigma_-^C(\hat{R})$ for the extended matrix $\hat{R}$ is obtained as the square root of the minimum eigenvalue of the matrix B_-^C given above in (2.2), where "$ext \equiv \min \equiv -$". Clearly, B_-^C has the following L^TL decomposition:

$$B_-^C = \begin{bmatrix} \sigma_-^C(R)^2 + (y_-^T v)^2 & \gamma(y_-^T v) \\ \gamma(y_-^T v) & \gamma^2 \end{bmatrix} = \begin{bmatrix} \sigma_-^C(R) & y_-^T v \\ 0 & \gamma \end{bmatrix} \begin{bmatrix} \sigma_-^C(R) & 0 \\ y_-^T v & \gamma \end{bmatrix} \equiv (L_-^C)^T L_-^C.$$

Further, the estimate $1/\sigma_+^C(\hat{R}^{-1})$ for

$$\hat{R}^{-1} = \begin{bmatrix} R^{-1} & -R^{-1}v/\gamma \\ 0 & 1/\gamma \end{bmatrix}$$

is the square root of $1/\lambda_+(B_+^C)$, where B_+^C is defined with respect to $\hat{R}^{-1}$. This value is also equal to the square root of $\lambda_-((B_+^C)^{-1})$. Using the assumptions (3.2) and (3.3), from (2.2) we subsequently get

$$\begin{aligned}
(B_+^C)^{-1} &= \begin{bmatrix} \left(\sigma_+^C(R^{-1})\right)^2 + (-x_+^T R^{-1}v/\gamma)^2 & -(x_+^T R^{-1}v)/\gamma^2 \\ -(x_+^T R^{-1}v)/\gamma^2 & 1/\gamma^2 \end{bmatrix}^{-1} \\
&= \begin{bmatrix} 1/(\sigma_-^C(R))^2 + ((y_-^T v)^2/(\sigma_-^C(R))^2\gamma^2) & -y_-^T v/(\sigma_-^C(R)\gamma^2) \\ -y_-^T v/(\sigma_-^C(R)\gamma^2) & 1/\gamma^2 \end{bmatrix}^{-1} \\
&= \left(\begin{bmatrix} 1/\sigma_-^C(R) & -y_-^T v/(\sigma_-^C(R)\gamma) \\ 0 & 1/\gamma \end{bmatrix} \begin{bmatrix} 1/\sigma_-^C(R) & 0 \\ -y_-^T v/(\sigma_-^C(R)\gamma) & 1/\gamma \end{bmatrix} \right)^{-1} \\
&= \begin{bmatrix} \sigma_-^C(R) & 0 \\ y_-^T v & \gamma \end{bmatrix} \begin{bmatrix} \sigma_-^C(R) & y_-^T v \\ 0 & \gamma \end{bmatrix}.
\end{aligned}$$

Clearly, we obtained the LL^T decomposition $(B_+^C)^{-1} = L_+^C(L_+^C)^T$, where L_+^C is the same as the factor L of the L^TL decomposition of B_-^C. That is, we have $L \equiv L_+^C = L_-^C$. It is easy to see from the SVD $U_L S W_L^T$ of L that B_+^C and $(B_-^C)^{-1}$ have the same eigenvalues. This implies the first part (3.2) of the theorem.

The approximate singular vectors for the extended problems are

$$\hat{y}_- = \begin{bmatrix} s_- y_- \\ c_- \end{bmatrix}, \qquad \hat{x}_+ = \begin{bmatrix} s_+ x_+ \\ c_+ \end{bmatrix},$$

where $[s_-, \quad c_-]^T$ is the eigenvector of $B_-^C = (L_-^C)^T L_-^C$ corresponding to its minimum eigenvalue and $[s_+, \quad c_+]^T$ is the eigenvector of $B_+^C = (L_+^C)^{-T}(L_+^C)^{-1}$ corresponding to its maximum eigenvalue. Then $[s_-, \quad c_-]^T = W_L e_2$ is also the right singular vector of L_-^C with the singular value $\sigma_-^C(\hat{R})$. Similarly, $[s_+, \quad c_+]^T$ is equal to $U_L e_2$ and it is also the right singular vector of $(L_+^C)^{-1}$ with the singular value $\sigma_+^C(\hat{R}^{-1}) = 1/\sigma_-^C(\hat{R})$. Taking all these into account, we get

$$\begin{aligned}
\hat{y}_-^T \hat{R} &= \begin{bmatrix} s_- y_-^T, & c_- \end{bmatrix} \begin{bmatrix} R & v \\ 0 & \gamma \end{bmatrix} = \begin{bmatrix} s_- y_-^T R, & s_- y_-^T v + c_- \gamma \end{bmatrix} = \begin{bmatrix} \sigma_-^C(R) s_- x_+^T, & s_- y_-^T v + c_- \gamma \end{bmatrix} \\
&= \begin{bmatrix} \sigma_-^C(R) s_-, & s_- y_-^T v + c_- \gamma \end{bmatrix} \begin{bmatrix} x_+^T & \\ & 1 \end{bmatrix} = \begin{bmatrix} s_-, & c_- \end{bmatrix} (L_-^C)^T \begin{bmatrix} x_+^T & \\ & 1 \end{bmatrix} \\
&= e_2^T W_L^T W_L S U_L^T \begin{bmatrix} x_+^T & \\ & 1 \end{bmatrix} = \sigma_-^C(\hat{R}) e_2^T U_L^T \begin{bmatrix} x_+^T & \\ & 1 \end{bmatrix} = \sigma_-^C(\hat{R}) \hat{x}_+^T.
\end{aligned}$$

We remark that the previous equalities also hold in the special case where B_-^C has two identical eigenvalues and where ICE defines $\begin{bmatrix} s_-, & c_- \end{bmatrix}^T = e_2^T$ and $\begin{bmatrix} s_+, & c_+ \end{bmatrix}^T = e_2^T$. ∎

Note that we can prove analogously that $\sigma_+^C(R) = 1/\sigma_-^C(R^{-1})$. Hence the ICE estimate of the condition number of R is always identical with the reciprocal of the ICE estimate of the condition number of R^{-1}. Now let us consider the alternative INE technique. INE deals with the *right* singular vectors of a triangular matrix. The following lemma is just an analogue of Lemma 3.1 for right singular vectors.

LEMMA 3.3. *Let R be a nonsingular matrix. Then the extremal singular values of R and R^{-1} satisfy $\sigma_-(R) = 1/\sigma_+(R^{-1})$. The corresponding right singular vectors z_- and x_+ of R and R^{-1}, respectively, satisfy*

$$\sigma_-(R) x_+ = R z_-. \tag{3.4}$$

Proof. As above, the first part is trivial. Let $R = USW^T$ be the SVD of R with the singular values in S in nonascending order. Clearly, $z_- = We_n$. Since $R^{-1} = WS^{-1}U^T$ we also have $x_+ = Ue_n$. Furthermore, $R^{-1}Ue_n = WS^{-1}U^T Ue_n = 1/(\sigma_-(R))We_n$. This immediately implies (3.4). ∎

The following theorem shows that INE is inherently different from ICE and it reveals that there is no analogy with Theorem 3.2. In particular, Theorem 3.4 cannot be applied recursively for leading principal submatrices of growing dimension because the assumption $1/\sigma_+^N(R^{-1}) = \sigma_-^N(R)$ will in general cease to hold.

THEOREM 3.4. *Let R be a nonsingular upper triangular matrix. Assume that the INE estimates of the singular values of R and R^{-1} satisfy $1/\sigma_+^N(R^{-1}) = \sigma_-^N(R) = \sigma_-(R)$. Then the INE estimates of the singular values related to the extended matrix* (2.1) *satisfy*

$$1/\sigma_+^N(\hat{R}^{-1}) \le \sigma_-^N(\hat{R})$$

with equality if and only if v in (2.1) *is collinear with the left singular vector corresponding to the smallest singular value of R.*

Proof. Consider the INE process applied to $\hat{R}$. The estimate $\sigma_-^N(\hat{R})$ is given by the square root of the minimum eigenvalue of the matrix B_-^N obtained from (2.3) by setting "$ext \equiv \min \equiv -$", which is also equal to the inverse of the square root of the maximum eigenvalue of the matrix $(B_-^N)^{-1}$. The $L^T L$ decomposition of the matrix $(B_-^N)^{-1}$ is derived as follows using Lemma 3.3 and its notation:

$$(B_-^N)^{-1} = \begin{bmatrix} z_-^T R^T R z_- & v^T R z_- \\ v^T R z_- & v^T v + \gamma^2 \end{bmatrix}^{-1} = \begin{bmatrix} \sigma_-(R)^2 & \sigma_-(R) v^T x_+ \\ \sigma_-(R) v^T x_+ & v^T v + \gamma^2 \end{bmatrix}^{-1}$$
$$= \left(\begin{bmatrix} \sigma_-(R) & 0 \\ v^T x_+ & \sqrt{v^T v - (v^T x_+)^2 + \gamma^2} \end{bmatrix} \begin{bmatrix} \sigma_-(R) & v^T x_+ \\ 0 & \sqrt{v^T v - (v^T x_+)^2 + \gamma^2} \end{bmatrix} \right)^{-1}$$
$$= L_-^T L_-$$

with

$$L_- = \begin{bmatrix} 1/\sigma_-(R) & 0 \\ -v^T x_+ / \left(\sigma_-(R) \sqrt{v^T v - (v^T x_+)^2 + \gamma^2} \right) & 1/\sqrt{v^T v - (v^T x_+)^2 + \gamma^2} \end{bmatrix}.$$

Further, the INE estimate for $1/\sigma_+^N(\hat{R}^{-1})$ is obtained from the eigenvalues of the matrix B_+^N which can be put down and represented in the form of an LL^T decomposition. Its derivation uses the fact that $\sigma_-(R) R^{-T} z_- = x_+$, which can be easily seen from the SVD $R = USW^T$ with $z_- = We_n$ and $x_+ = Ue_n$. Then with Lemma 3.3, $R^{-T}R^{-1}x_+ = x_+/\sigma_-(R)^2$. A few simple steps provide

$$B_+^N = \begin{bmatrix} x_+^T R^{-T} R^{-1} x_+ & -x_+^T R^{-T} R^{-1} v/\gamma \\ -x_+^T R^{-T} R^{-1} v/\gamma & v^T R^{-T} R^{-1} v/\gamma^2 + 1/\gamma^2 \end{bmatrix}$$
$$= \begin{bmatrix} 1/\sigma_-(R)^2 & -v^T x_+/(\sigma_-(R)^2 \gamma) \\ -v^T x_+/(\sigma_-(R)^2 \gamma) & (\|R^{-1}v\|^2 + 1)/\gamma^2 \end{bmatrix} = L_+ L_+^T$$

with

$$L_+ = \begin{bmatrix} 1/\sigma_-(R) & 0 \\ -v^T x_+/(\sigma_-(R)\gamma) & \left(\sqrt{\|R^{-1}v\|^2 - (v^T x_+)^2/\sigma_-(R)^2 + 1} \right)/\gamma \end{bmatrix}.$$

The Cauchy inequality $(v^T x_+)^2 \le v^T v$ and properties of the SVD imply

$$\|R^{-1}v\|^2 = \|S^{-1}U^T v\|^2 = \sum_{j=1}^{n} \frac{(e_j^T U^T v)^2}{s_{jj}^2} \ge (v^T x_+)^2/\sigma_-(R)^2. \tag{3.5}$$

This implies the relation

$$\|L_-\| = \left\| \begin{bmatrix} 1 & \frac{\gamma}{\sqrt{v^T v - (v^T x_+)^2 + \gamma^2}} \end{bmatrix} L_+ \begin{bmatrix} 1 & \frac{1}{\sqrt{\|R^{-1}v\|^2 - (v^T x_+)^2/\sigma_-^N(R)^2 + 1}} \end{bmatrix} \right\| \le \|L_+\|.$$

The involved norms of the triangular factors directly provide

$$\left(\sigma_+^N(\hat{R}^{-1}) \right)^{-1} = \|L_+\|^{-1} \le \|L_-\|^{-1} = \sigma_-^N(\hat{R}). \tag{3.6}$$

Equality in (3.6) is attained if and only if $(v^T x_+)^2/(\sigma_-(R))^2 = \|R^{-1}v\|^2$ and also $(v^T x_+)^2 = v^T v$. These two conditions are equivalent with the collinearity of v with $x_+ = Ue_n$. □

We can obtain the analogue result for the approximate *largest* singular value $\sigma_+^N(\hat{R})$ if we consider in Theorem 3.4 instead of $\hat{R}$ its inverse. Let us denote the inverse of $\hat{R}$ by $\hat{S}$. Using Theorem 3.4 we get $\sigma_-^N(\hat{R}) = \sigma_-^N(\hat{S}^{-1}) \ge 1/\sigma_+^N(\hat{R}^{-1}) = 1/\sigma_+^N(\hat{S})$,

i.e., for any upper triangular S with $1/\sigma_-^N(S^{-1}) = \sigma_+^N(S) = \sigma_+(S)$ we have for the extended matrix $\hat{S}$

$$\sigma_+^N(\hat{S}) \geq 1/\sigma_-^N(\hat{S}^{-1}). \tag{3.7}$$

Consequently, under the assumption of starting with exact estimates like in Theorem 3.4, INE will be more accurate when estimating σ_-, respectively, σ_+, if one applies incremental *maximization* (using $1/\sigma_+^N$ or σ_+^N, respectively) instead of incremental *minimization* (using σ_-^N or $1/\sigma_-^N$, respectively). This is in contrast with the ICE technique, where maximization and minimization give identical approximations in the sense of (3.2). When the inverse is not available, Theorem 3.4 and (3.7) seem to suggest that the quality of the INE estimate of the largest singular value might in most cases be better than the quality of the estimate for the smallest singular value. Further, Theorem 3.4 and (3.7) assume that the INE estimates of the singular values of R and R^{-1} are exact. Our experiments suggest that even in the more general situation when the assumptions of Theorem 3.4 may not hold, minimization works better than maximization very rarely in practice. In fact, in our tests with various types of matrices traditionally used to assess the quality of incremental condition estimators and with matrices from the Matrix Market collection [17] this never occurred. In order to better understand this behavior, we propose to consider the following expressions for $1/\sigma_+^N(\hat{R}^{-1})$ and $\sigma_-^N(\hat{R})$.

PROPOSITION 3.5. *Let R be a nonsingular upper triangular matrix and let the INE approximate singular vectors for $\sigma_+^N(R^{-1})$ and $\sigma_-^N(R)$ be denoted by x_+ and z_-, respectively. Then the INE estimates of the singular values related to the extended matrix* (2.1) *satisfy*

$$\sigma_-^N(\hat{R}) = \sigma_-(L_-^N), \qquad L_-^N = \begin{bmatrix} \sigma_-^N(R) & 0 \\ \iota_- & \sqrt{\gamma^2 + v^T v - \iota_-^2} \end{bmatrix}, \quad \iota_- = v^T R z_- / \sigma_-^N(R)$$

and

$$1/\sigma_+^N(\hat{R}^{-1}) = \sigma_-(L_+^N), \qquad L_+^N = \begin{bmatrix} 1/\sigma_+^N(R^{-1}) & 0 \\ \dfrac{\iota_+}{\sqrt{\|R^{-1}v\|^2 - (\frac{\iota_+}{\sigma_+})^2 + 1}} & \dfrac{\gamma}{\sqrt{\|R^{-1}v\|^2 - (\frac{\iota_+}{\sigma_+})^2 + 1}} \end{bmatrix},$$

where $\sigma_+ = \sigma_+^N(R^{-1})$, $\iota_+ = v^T R^{-T} R^{-1} x_+ / \sigma_+^2$.

Proof. The estimate $\sigma_-^N(\hat{R})$ is given by the root of the minimum eigenvalue of the matrix B_-^N obtained from (2.3) by setting "$ext \equiv \min \equiv -$". The Cholesky decomposition of the matrix B_-^N is

$$\begin{aligned} B_-^N &= \begin{bmatrix} z_-^T R^T R z_- & v^T R z_- \\ v^T R z_- & v^T v + \gamma^2 \end{bmatrix} \\ &= \begin{bmatrix} \sigma_-^N(R) & 0 \\ \iota_- & \sqrt{\gamma^2 + v^T v - \iota_-^2} \end{bmatrix} \begin{bmatrix} \sigma_-^N(R) & \iota_- \\ 0 & \sqrt{\gamma^2 + v^T v - \iota_-^2} \end{bmatrix} = L_-^N (L_-^N)^T. \end{aligned}$$

This gives $\sigma_-^N(\hat{R}) = \sigma_-(L_-^N)$. Similarly, the estimate $1/\sigma_+(\hat{R}^{-1})^N$ is given by the root of the minimum eigenvalue of the matrix $(B_+^N)^{-1}$ obtained from (2.3) and defined with respect to $\hat{R}^{-1}$ by setting "$ext \equiv \max \equiv +$". The $L^T L$ decomposition of the matrix $(B_+^N)^{-1}$ is

$$
\begin{aligned}
(B_+^N)^{-1} &= \begin{bmatrix} x_+^T R^{-T} R^{-1} x_+ & -v^T R^{-T} R^{-1} x_+/\gamma \\ -v^T R^{-T} R^{-1} x_+/\gamma & v^T R^{-T} R^{-1} v/\gamma^2 + 1/\gamma^2 \end{bmatrix}^{-1} \\
&= \begin{bmatrix} \sigma_+^N(R^{-1})^2 & -\iota_+ \sigma_+^N (R^{-1})^2/\gamma \\ -\iota_+ \sigma_+^N (R^{-1})^2/\gamma & \|R^{-1}v\|^2/\gamma^2 + 1/\gamma^2 \end{bmatrix}^{-1} \\
&= \begin{bmatrix} \sigma_+^N(R^{-1}) & -\iota_+ \sigma_+^N (R^{-1})/\gamma \\ 0 & \sqrt{||R^{-1}v||^2 - \iota_+^2 \sigma_+^N (R^{-1})^2 + 1}/\gamma \end{bmatrix}^{-1} \\
&= \begin{bmatrix} \sigma_+^N(R^{-1}) & 0 \\ -\iota_+ \sigma_+^N (R^{-1})/\gamma & \sqrt{||R^{-1}v||^2 - \iota_+^2 \sigma_+^N (R^{-1})^2 + 1}/\gamma \end{bmatrix}^{-1} = (L_+^N)^T L_+^N.
\end{aligned}
$$

The claim follows from

$$
L_+^N = \begin{bmatrix} 1/\sigma_+^N(R^{-1}) & 0 \\ \iota_+/\sqrt{||R^{-1}v||^2 - \iota_+^2/(\sigma_+^N(R^{-1}))^2 + 1} & \gamma/\sqrt{||R^{-1}v||^2 - \iota_+^2/(\sigma_+^N(R^{-1}))^2 + 1} \end{bmatrix}. \qquad \square
$$

For a partial explanation of why maximization seems in general to outperform minimization, let us compare the entries of the matrices L_-^N and L_+^N defined in Proposition 3.5. Since we have $i_-^2 \le v^T v$ and $\iota_+^2/(\sigma_+^N(R^{-1}))^2 \le ||R^{-1}v||^2$, the second diagonal entry of L_+^N is always smaller than that of L_-^N. When the dimension of $\hat{R}$ is two, the first diagonal entries of L_-^N are L_+^N identical at the beginning of the estimation process, because they are exact. When $\hat{R}$ has dimension three, the first diagonal entry of L_+^N is not larger than that of L_-^N from Theorem 3.4. Further, when started with $1/\sigma_+^N(R^{-1}) \le \sigma_-^N(R)$, in order for $1/\sigma_+^N(\hat{R}^{-1}) \le \sigma_-^N(\hat{R})$ to hold it clearly suffices that the off-diagonal entries of L_+^N and L_-^N satisfy the simple inequality stated in the following corollary.

COROLLARY 3.6. *Using the notation of Proposition* 3.5 *and assuming* $1/\sigma_+^N(R^{-1}) \le \sigma_-^N(R)$*, there holds*

$$
1/\sigma_+^N(\hat{R}^{-1}) \le \sigma_-^N(\hat{R}) \qquad \textit{if} \qquad |\iota_-| \le \left| \frac{\iota_+}{\sqrt{\|R^{-1}v\|^2 - (\frac{\iota_+}{\sigma_+})^2 + 1}} \right|. \tag{3.8}
$$

The following example shows that the sufficient condition in the previous corollary may possibly be simplified but it cannot be removed. Let us consider matrices R and R^{-1} defined as follows:

$$
R = \begin{bmatrix} 2 & 0 & 1 \\ & 1 & 0 \\ & & 1 \end{bmatrix}, \qquad R^{-1} = \begin{bmatrix} \frac{1}{2} & 0 & -\frac{1}{2} \\ & 1 & 0 \\ & & 1 \end{bmatrix}.
$$

The ICE estimate $\sigma_-^C(R)$ for the smallest singular value $\sigma_-(R) = 0.874$ is $\sigma_-^C(R) = 1$. The ICE estimate $1/\sigma_+^C(R^{-1})$ is of the same value, i.e., $1/\sigma_+^C(R^{-1}) = 1$, which is in agreement with Theorem 3.2. Note that here we used a matrix in block angular form that does not pass the information in ICE as discussed in [5]. The INE estimate $\sigma_-^N(R)$ for the smallest singular value is also $\sigma_-^N(R) = 1$, but the INE estimate $1/\sigma_+^N(R^{-1})$ is more accurate since $1/\sigma_+^N(R^{-1}) = \sqrt{4/5} \approx 0.8944$. This is what one would expect from Theorem 3.4. (Its assumptions are satisfied because the estimates for triangular matrices of size two are always exact.)

Consider now an extended matrix $\hat{R}$ with $\gamma = 1$ in (2.1). The choice of v influences the values ι_- and ι_+ in Proposition 3.5, which can be crucial for whether $1/\sigma_+^N(\hat{R}^{-1}) < \sigma_-^N(\hat{R})$ holds; see Corollary 3.6. The INE approximation of the right singular vector z_- corresponding to $\sigma_-^N(R)$ is $z_- = [0,1,0]^T$, hence $\iota_- = (v^T R z_-)/\sigma_-^N(R) = v^T[0,1,0]^T$. Similarly, using the INE approximate right singular vector $x_+ = [0,0,1]^T$ corresponding to $1/\sigma_+^N(R^{-1})$ we arrive at $\iota_+ = (v^T R^{-T} R^{-1} x_+)/\sigma_+^N(R^{-1})^2 = 4/5 \cdot v^T[-1/4, 0, 5/4]^T$. Let us consider the vector $v = [1,1,1]^T$ giving

$$\hat{R} = \begin{bmatrix} 2 & 0 & 1 & 1 \\ & 1 & 0 & 1 \\ & & 1 & 1 \\ & & & 1 \end{bmatrix}, \quad \sigma_-(\hat{R}) \approx 0.5155, \quad \iota_- = 1, \quad \iota_+ = 4/5, \qquad \text{and}$$

$$0.5381 \approx \left(\frac{17/4 + \sqrt{(17/4)^2 - 11}}{2} \right)^{-\frac{1}{2}} = 1/\sigma_+^N(\hat{R}^{-1}) < \sigma_-^N(\hat{R}) = \sqrt{\frac{5-\sqrt{13}}{2}} \approx 0.835,$$

which is what one may expect from Proposition 3.5. Just note that the ICE estimates are

$$\sigma_-^C(\hat{R}) = 1/\sigma_+^C(\hat{R}^{-1}) = \sqrt{\frac{3-\sqrt{5}}{2}} \approx 0.618.$$

We can, however, construct a case where the sufficient condition of Corollary 3.6 is not satisfied and $1/\sigma_+^N(\hat{R}^{-1}) > \sigma_-^N(\hat{R})$ by making ι_+ smaller. For instance, with $v = [0,1,0]^T$ we have $\iota_+ = 0$ and $\iota_- = 1$. The extended matrix is then

$$\hat{R} = \begin{bmatrix} 2 & 0 & 1 & 0 \\ & 1 & 0 & 1 \\ & & 1 & 0 \\ & & & 1 \end{bmatrix} \qquad \text{with} \quad \sigma_-(\hat{R}) = \sqrt{\frac{3-\sqrt{5}}{2}},$$

and we obtain

$$0.618 \approx \sqrt{\frac{3-\sqrt{5}}{2}} = \sigma_-^N(\hat{R}) < 1/\sigma_+^N(\hat{R}^{-1}) = \sqrt{\frac{1}{2}} \approx 0.7071.$$

The ICE estimates satisfy in this case $\sigma_-^C(\hat{R}) = 1/\sigma_+^C(\hat{R}^{-1}) = 1$.

This example might indicate that it is not too difficult to find academic examples where estimating $\sigma_-(\hat{R})$ by $\sigma_-^N(\hat{R})$ (i.e., with minimization) works better than using $1/\sigma_+^N(\hat{R}^{-1})$ (i.e., maximization). But as we mentioned, we never observed this in practice. Let us give one striking example. In Figure 3.1 the pluses display the minimum singular value of the one-dimensional Laplacians $\mathcal{L}_i$, $i = 1, \ldots, 100$, of size 1 to 100. The circles represent the INE estimates $1/\sigma_+^N(\mathcal{L}_i^{-1})$, $i = 1, \ldots, 100$, and they are very accurate. (See also Figure 3.2, which is a zoom of Figure 3.1 for the INE estimates $1/\sigma_+^N(\mathcal{L}_i^{-1})$, $i = 50, \ldots, 100$.) The solid line represents the INE estimates $\sigma_-^N(\mathcal{L}_i)$, $i = 1, \ldots, 100$ based on minimization. They stagnate around the value 0.6356.

Summarizing, we present at the end of this section a number of results suggesting superiority of INE maximization based on the inverse of the triangular factor over INE minimization. A sound theoretical explanation for this phenomenon, which is often observed but for which counterexamples can be constructed (see above), is an open problem.

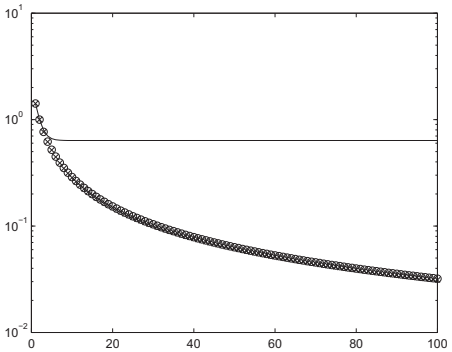


FIG. 3.1. *INE estimation of the smallest singular value of the one-dimensional Laplacians of size 1 to 100: INE with minimization (solid line), INE with maximization (circles), and exact minimum singular values (pluses).*

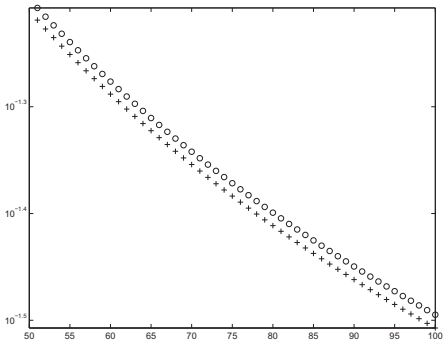


FIG. 3.2. *INE estimation of the smallest singular value of the one-dimensional Laplacians of size 50 to 100, zoom of Figure* 3.1 *for INE with maximization and exact minimum singular values.*

4. Superiority of INE maximization over ICE maximization. While the previous section concludes that the maximization problem in INE should be preferred for estimating both the maximum and the minimum singular value (exploiting the inverse), this section addresses the question of whether the ICE technique can be more efficient than INE when the inverse is available. We already proved that using the inverse does not improve the ICE technique (Theorem 3.2), but this does not mean that ICE estimates are worse than INE estimates exploiting the inverse. If ICE maximization were more powerful than INE maximization, there would hold, with the assumptions of Theorem 3.4,

$$\sigma_-^C(\hat{R}) = 1/\sigma_+^C(\hat{R}^{-1}) \leq 1/\sigma_+^N(\hat{R}^{-1}) \leq \sigma_-^N(\hat{R})$$

and in that case also ICE *minimization* would be more powerful than INE minimization. We therefore concentrate on maximization. The subsequent text presents sufficient conditions for the opposite case, that is, for the superiority of INE maximization to ICE maximization. Extensive numerical experiments confirm that INE maximization is the method of choice. We also graphically demonstrate the strength of the introduced sufficient conditions. Let us compare INE and ICE maximization from the theoretical point of view first.

Similarly to the results of the previous section, we are not able to prove the superiority of INE unconditionally and counterexamples exist. This type of conclusion seems to be present in many areas connected with condition estimators that can sometimes fail. On the other hand we are just interested in proposing a strategy which would give results as good as possible on average and we believe that we are successful in this. We will see that both the theoretical arguments, the figures displayed in this section, and also the results in the experimental section support the claim that INE maximization is preferable over ICE maximization.

The theoretical arguments consist of the two following theorems that provide sufficient conditions for superiority of INE.

THEOREM 4.1. *Consider norm estimation of the extended matrix* (2.1) *where ICE and INE start with the same approximation* $\sigma_+ \equiv \sigma_+^C(R) = \sigma_+^N(R)$. *Let* y *be the corresponding approximate left singular vector, let* z *be the corresponding approximate*

right singular vector, and let $w = Rz/\sigma_+$. *Then the approximation* $\sigma_+^N(\hat{R})$ *obtained from INE is at least as good as the approximation* $\sigma_+^C(\hat{R})$ *from ICE if*

$$(v^T w)^2 \geq (v^T y)^2. \tag{4.1}$$

Proof. The largest eigenvalue of B_+^C from (2.2) (with the simplified notation introduced here) corresponds to the rightmost intersection of the parabola $\ell(\lambda) = \left(\lambda - \sigma_+^2 - (v^T y)^2\right)(\lambda - \gamma^2)$ with the horizontal line $h(\lambda) \equiv \gamma^2 (v^T y)^2$. Hence the largest eigenvalue λ_R of B_+^N from (2.3) is larger than or equal to the leading eigenvalue of B_+^C if and only if

$$\ell(\lambda_R) \geq \gamma^2 (v^T y)^2. \tag{4.2}$$

The condition (4.2) corresponds to the case when INE maximization for $\hat{R}$ is at least as good as ICE maximization for the same matrix. Substituting

$$\lambda_R \equiv \frac{1}{2}\left(\sigma_+^2 + v^T v + \gamma^2 + S\right), \qquad S \equiv \sqrt{(\sigma_+^2 - \gamma^2 - v^T v)^2 + 4\sigma_+^2 (v^T w)^2} \tag{4.3}$$

into $\ell(\lambda_R)$ we have

$$\begin{aligned}
\ell(\lambda_R) &= \left(\gamma^2 - \lambda_R\right)\left(\sigma_+^2 + (v^T y)^2 - \lambda_R\right) \\
&= \frac{1}{4}\left(\gamma^2 - \sigma_+^2 - v^T v - S\right)\left(\sigma_+^2 + 2(v^T y)^2 - v^T v - \gamma^2 - S\right) \\
&= \frac{1}{4}\left((\gamma^2 - \sigma_+^2 - v^T v)(\sigma_+^2 - \gamma^2 - v^T v + 2(v^T y)^2) - 2((v^T y)^2 - v^T v)S + S^2\right).
\end{aligned}$$

Thus (4.2) is satisfied if and only if

$$(\gamma^2 - \sigma_+^2 - v^T v)(\sigma_+^2 - \gamma^2 - v^T v + 2(v^T y)^2) - 2((v^T y)^2 - v^T v)S + S^2 \;\geq\; 4\gamma^2 (y^T v)^2.$$

Substituting S^2 from (4.3) we can obtain

$$2\left(\sigma_+^2 - \gamma^2 + v^T v + S + 2\gamma^2\right)\left(v^T v - (v^T y)^2\right) - 4\sigma_+^2\left(v^T v - (v^T w)^2\right) \;\geq\; 0,$$

and after some rewriting we arrive at the equivalent condition

$$2\left(\gamma^2 - \sigma_+^2 + v^T v + S\right)\left(v^T v - (v^T y)^2\right) + 4\sigma_+^2\left((v^T w)^2 - (v^T y)^2\right) \geq 0 \tag{4.4}$$

that is equivalent to (4.2). The Cauchy inequality implies that $v^T v - (v^T y)^2 \geq 0$. If $v^T v - (v^T y)^2 = 0$, then we are done since (4.4) follows directly from (4.1).

Consider $v^T v - (v^T y)^2 > 0$. Let $\epsilon \geq 0$ be defined through

$$(v^T w)^2 - (v^T y)^2 = \epsilon\left(v^T v - (v^T y)^2\right). \tag{4.5}$$

Then (4.4) implies that (4.2) is satisfied if and only if

$$2\left(\gamma^2 + v^T v + S + (2\epsilon - 1)\sigma_+^2\right) \geq 0, \tag{4.6}$$

that is, if and only if

$$S^2 = (\sigma_+^2 - \gamma^2 - v^T v)^2 + 4\sigma_+^2 (v^T w)^2 \geq \left(\sigma_+^2 - \gamma^2 - v^T v - 2\epsilon\sigma_+^2\right)^2.$$

Equivalently, (4.2) is valid with $v^Tv - (v^Ty)^2 > 0$ if and only if

$$\epsilon^2\sigma_+^2 - \epsilon(\sigma_+^2 - \gamma^2 - v^Tv) - (v^Tw)^2 \leq 0. \tag{4.7}$$

This is true for $\epsilon = 0$. But this means, in view of (4.6), that for $\epsilon = 0$

$$\gamma^2 + v^Tv + S + (2\epsilon - 1)\delta^2 \geq 0.$$

Consequently, for *all* $\epsilon \geq 0$,

$$\gamma^2 + v^Tv + S + (2\epsilon - 1)\delta^2 \geq 0. \qquad \square$$

The next theorem formulates an even stricter sufficient condition for the superiority of INE. This condition seems to be rather technical but it enables us to specify more precisely the areas of parameters where one of the techniques is better than the other. We will see that based on the input parameters of the condition estimator, there is always a possibility that the INE technique is better than ICE, but not vice versa.

THEOREM 4.2. *Using the same notation and assumptions as in Theorem* 4.1*, the approximation* $\sigma_+^N(\hat{R})$ *obtained from INE is at least as good as the approximation* $\sigma_+^C(\hat{R})$ *from ICE if*

$$(v^Tw)^2 \geq \rho_1, \tag{4.8}$$

where ρ_1 *is the smaller root of the quadratic equation in* $(v^Tw)^2$*,*

$$\begin{aligned}(v^Tw)^4 + \left(\frac{\gamma^2 + (v^Ty)^2}{\sigma_+^2}\left(v^Tv - (v^Ty)^2\right) - v^Tv - (v^Ty)^2\right)(v^Tw)^2 \\ + (v^Ty)^2\left(\frac{\gamma^2 + v^Tv}{\sigma_+^2}\left((v^Ty)^2 - v^Tv\right) + v^Tv\right) = 0.\end{aligned} \tag{4.9}$$

Proof. Assume for the moment that $v^Tv - (v^Ty)^2 > 0$. Let us substitute the expression for ϵ from (4.5) into the inequality (4.7). We get directly

$$\begin{aligned}&\left(\frac{(v^Tw)^2 - (v^Ty)^2}{v^Tv - (v^Ty)^2}\right)^2\sigma_+^2 - \frac{\left((v^Tw)^2 - (v^Ty)^2\right)\left(v^Tv - (v^Ty)^2\right)}{\left(v^Tv - (v^Ty)^2\right)^2}(\sigma_+^2 - \gamma^2 - v^Tv) \\ &\quad - \frac{(v^Tw)^2\left(v^Tv - (v^Ty)^2\right)^2}{\left(v^Tv - (v^Ty)^2\right)^2} \leq 0,\end{aligned}$$

and after a few simple steps we obtain the sufficient condition for the superiority of INE

$$\rho_1 \leq (v^Tw)^2 \leq \rho_2, \tag{4.10}$$

where $\rho_{1,2}$ are the roots of (4.9). They have the form

$$(v^Ty)^2 + \frac{\left(v^Tv - (v^Ty)^2\right)}{2\sigma_+^2}\left(\beta \pm \sqrt{\beta^2 + 4\sigma_+^2(v^Ty)^2}\right), \tag{4.11}$$

where $\beta = \sigma_+^2 - \gamma^2 - (v^Ty)^2$. Clearly, we get

$$\rho_1 \leq (v^Ty)^2 \leq \rho_2. \tag{4.12}$$

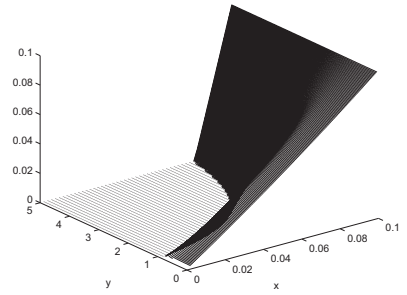


FIG. 4.1. *Value of* ρ_1 *in* (4.8) *in dependence of* $(v^T y)^2$ *(x-axis) and* γ^2 *(y-axis) with* $v^T v = 0.1$, $\sigma_+ = 1$ *and with* $\Delta = 0$ *in* (4.13).

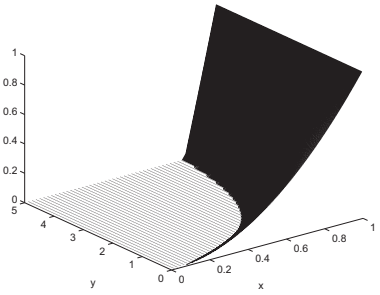


FIG. 4.2. *Value of* ρ_1 *in* (4.8) *in dependence of* $(v^T y)^2$ *(x-axis) and* γ^2 *(y-axis) with* $v^T v = 1$, $\sigma_+ = 1$ *and with* $\Delta = 0$ *in* (4.13).

If $(v^T w)^2 < (v^T y)^2$, then (4.8) and (4.12) imply superiority of INE based on (4.10); otherwise Theorem 4.1 can be applied. Finally, if $v^T v - (v^T y)^2 = 0$, then the roots of (4.9) coincide and take the value $\rho_{1,2} = (v^T y)^2$; see (4.11). Hence the condition (4.8) reduces to (4.1) and again, Theorem 4.1 can be applied. □

An important conclusion of the previous theorems is as follows. Let us divide the possible input vectors v into two sets. The first set contains v such that $(v^T w)^2 \geq (v^T y)^2$ and the second set contains the other instances of v. Then, the sufficient condition for superiority is *always* valid for all v from the first group and it is possibly valid also for some v from the second group. In particular, INE is never worse than ICE under the assumptions of these theorems whenever $\rho_1 \leq 0$. We do not have a similar claim for the superiority of ICE.

4.1. Graphical demonstration. In this subsection we graphically demonstrate the relation between ICE and INE maximization that points out the superiority of the latter approach. The presented figures depict on the z-axis the value $\max(0, \rho_1)$, that is, the sufficient condition for the superiority of INE estimation in (4.8), where we display $\max(0, \rho_1)$ because for $\rho_1 \leq 0$ the condition is automatically satisfied. If we scale the matrix such that $\sigma_+ = 1$, and this can always be done without loss of generality, the coefficients of (4.9) depend on three variables only. These three variables are $(v^T y)^2$, $v^T v$, and γ^2. Fixing $v^T v$, we can display the dependence on the other variables in the remaining two dimensions of the figures. We plot the values of $(v^T y)^2$ on the x-axis and γ^2 on the y-axis. For practical reasons, we restrict ourselves to $\gamma^2 \leq 5$ but the behavior for larger values is more or less the same as for $\gamma^2 = 5$. Figures 4.1–4.3 display the values for three different choices of the norm $v^T v$. We know from Theorem 4.2 that INE is *unconditionally* (regardless of the vector w) superior over ICE for $\rho_1 \leq 0$. In our pictures this case corresponds to its crosshatched part. In the other cases (dark part of the figures), the conclusion of whether ICE or INE maximization is better still depends on the mutual relation of $(v^T w)^2$ and $(v^T y)^2$ and either of the techniques can be better than the other.

Let us mention here that a result similar to Theorem 4.2 could be derived that uses as an additional parameter the distance

$$\Delta \equiv \sqrt{(\sigma_+^N)^2 - (\sigma_+^C)^2}, \qquad \sigma_+^N \geq \sigma_+^C, \tag{4.13}$$

with $\sigma_+^N = \sigma_{max}^N(R)$ and $\sigma_+^C = \sigma_{max}^C(R)$. The previous case corresponds to the case $\Delta = 0$. The claims and proofs are very similar and we omit them here since they would

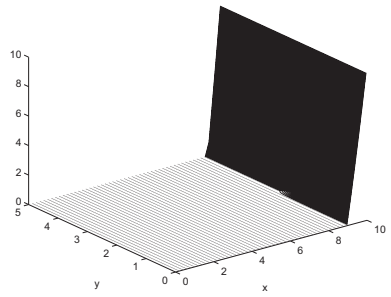


FIG. 4.3. *Value of* ρ_1 *in* (4.8) *in dependence of* $(v^T y)^2$ *(x-axis) and* γ^2 *(y-axis) with* $v^T v = 10$, $\sigma_+ = 1$ *and with* $\Delta = 0$ *in* (4.13).

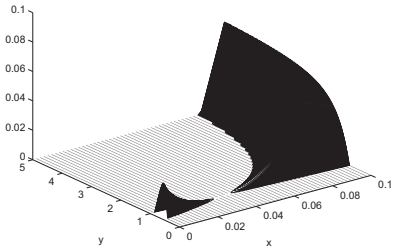


FIG. 4.4. *Value of* ρ_1 *in* (4.8) *in dependence of* $(v^T y)^2$ *(x-axis) and* γ^2 *(y-axis) with* $v^T v = 0.1$, $\sigma_+ = 1$ *and with* $\Delta = 0.6$ *in* (4.13).

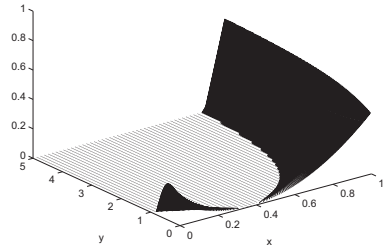


FIG. 4.5. *Value of* ρ_1 *in* (4.8) *in dependence of* $(v^T y)^2$ *(x-axis) and* γ^2 *(y-axis) with* $v^T v = 1$, $\sigma_+ = 1$ *and with* $\Delta = 0.6$ *in* (4.13).

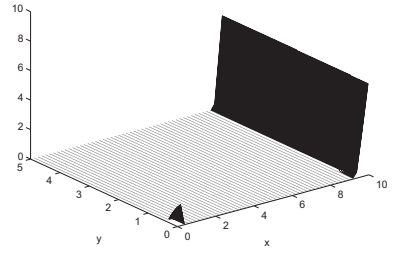


FIG. 4.6. *Value of* ρ_1 *in* (4.8) *in dependence of* $(v^T y)^2$ *(x-axis) and* γ^2 *(y-axis) with* $v^T v = 10$, $\sigma_+ = 1$ *and with* $\Delta = 0.6$ *in* (4.13).

not give additional insight for our statement that INE maximization is preferable over ICE maximization. Nevertheless, just for illustration, we present here figures for the same choices of values of $||v||$ and with nonzero Δ; here, $\Delta = 0.6$. In Figures 4.4–4.6 we can see that the results are as we would intuitively expect; $\Delta > 0$ seems even to increase the expectation for the superiority of INE over ICE.

Let us recall the one-dimensional Laplacian example from section 3. It shows not only that INE maximization based on the inverse matrix may be very accurate, but it also points out that the estimate of σ_-^N via INE minimization can be *very* poor. Therfore, if the plain ICE-based strategy is used without the matrix inverse to estimate both singular values, the condition number estimate is often better than if plain INE without inverse is used. In other words, experiments show that INE minimization is by far the weakest point of the two investigated strategies. The explanation of this observation is an interesting open problem.

5. Numerical experiments. In this section we focus on illustrating the theoretical results in sections 3 and 4. In particular, we confirm that using just maximization in INE seems to be a better strategy than using minimization as well. Further, we will see that ICE is clearly outperformed by INE using various matrix test sets. The experiments, all run in MATLAB, show that the availability of the inverse inside the decomposition is desirable, but, except for the last experiment, we compute the inverse separately with MATLAB's backslash command.

Our experiments compare the following four strategies:

1. The original ICE technique from [3] with the estimates defined as $\sigma_+^C(R)/\sigma_-^C(R)$.
2. The INE technique from [18] for estimating both the norm and the minimum singular value with the estimates defined by $\sigma_+^N(R)/\sigma_-^N(R)$. Although INE was originally proposed for norm estimation only, we refer to this estimator as the original INE.
3. The INE technique based on maximization only, which also uses the inverse R^{-1}, that is, estimates defined as $\sigma_+^N(R)\,\sigma_+^N(R^{-1})$.
4. The INE technique based on minimization only which uses the matrix inverse as well, that is, $\left(\sigma_-^N(R)\,\sigma_-^N(R^{-1})\right)^{-1}$.

Note that we do not display any results for the estimates $\sigma_-^C(R^{-1})/\sigma_+^C(R^{-1})$ since, as we proved in Theorem 3.2, they are identical with the original ICE estimates.

5.1. Example 1. Using the MATLAB command `A = rand(100,100) - rand(100,100)` we generated 50 matrices A of size 100, computed a column pivoting using `colamd`, and obtained an upper triangular factor R from the QR decomposition of the column permuted matrix A. This is the same type of experiment as in [3, section 4, Test 1]. The condition estimators were tested on R; see Figures 5.1 and 5.2. Note that for simplicity we refer here also to an experiment from Example 2. When omitting the column pivoting we get qualitatively the same picture.

We can see that the estimate $\sigma_+^N(R)\sigma_+^N(R^{-1})$, which uses maximizing INE processes only, performs the best by far. On the other hand, the estimate $(\sigma_-^N(R)\sigma_-^N(R^{-1}))^{-1}$, which uses minimizing INE processes only, performs very poorly. This supports experimentally the fact mentioned above that INE is powerful when maximizing and weak when minimizing. The ICE technique performs moderately (and it cannot be improved by exploiting the inverse) and the original INE technique performs even worse, again, because of the weak performance when estimating the minimum singular value.

It may be interesting to see a comparison between the theoretically derived sufficient conditions for the superiority of INE maximization over ICE maximization. Figures 5.3 and 5.4 display the fraction of cases in which the sufficient conditions for superiority of INE maximization (4.1), (4.8), and (3.8) are satisfied if this superiority is actually achieved. Note that the first two conditions refer to a comparison of INE and ICE and the third just relates INE maximization and minimization. Overall, in about half of the cases the conditions are satisfied and they represent a nonnegligible case in the estimation process. We also see verified the fact that condition (4.1) is weaker than (4.8), as mentioned in section 4.

5.2. Example 2. We generated 50 matrices of the form $A = U\Sigma V^T$ of size 100 with a prescribed condition number κ by choosing $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_{100})$ with

$$\sigma_k = \alpha^k, \quad 1 \le k \le 100, \qquad \text{where} \quad \alpha = \kappa^{-\frac{1}{99}}.$$

U and V are the Q factors of the QR factorizations of matrices B generated using the MATLAB command `B = rand(100,100) - rand(100,100)`. Then we computed a column pivoting with the `colamd` command and obtained an upper triangular factor R from the QR decomposition of the permuted A. This corresponds to the experiments in [3, section 4, Test 2] and [18, section 5, Table 5.4]. The condition estimators were tested on R; see Figures 5.2, 5.5, 5.6 for $\kappa = 10, 100, 1000$, respectively. When omitting the column pivoting we get qualitatively the same picture.

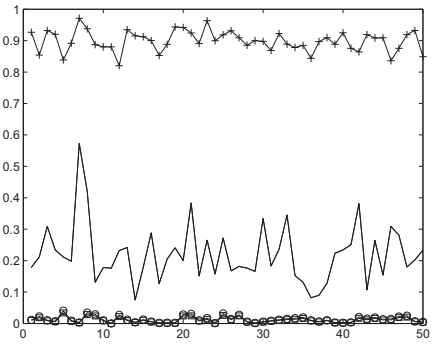


FIG. 5.1. *Ratio of estimate to real condition number for the* 50 *matrices in example* 1. *Solid line: ICE (original); pluses: INE with inverse and using only maximization; circles: INE (original); squares: INE with inverse and using only minimization.*

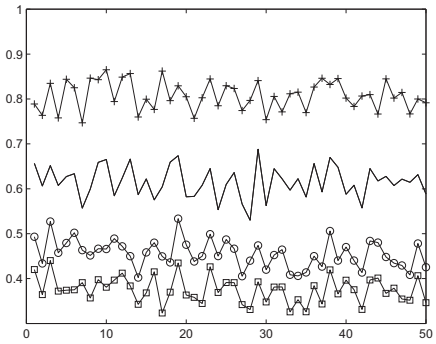


FIG. 5.2. *Ratio of estimate to real condition number for the matrices in example* 2 *with* $\kappa(A) = 10$. *Solid line: ICE (original); pluses: INE with inverse and using only maximization; circles: INE (original); squares: INE with inverse and using only minimization.*

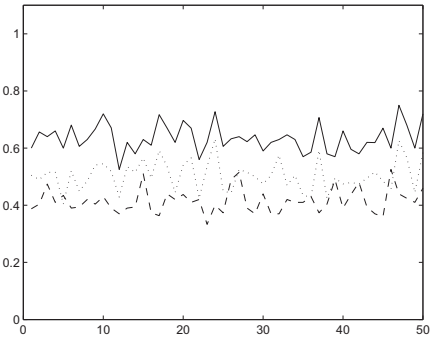


FIG. 5.3. *Ratio of the satisfied sufficient conditions in condition number estimation for the* 50 *matrices in example* 1. *Solid line:* (4.8)*; dotted:* (4.1)*; dashed:* (3.8).

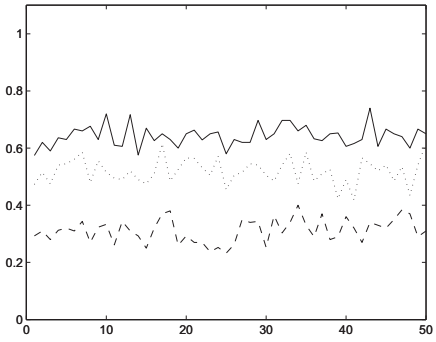


FIG. 5.4. *Ratio of the satisfied sufficient conditions in condition number estimation for the* 50 *matrices in example* 2 *with* $\kappa(A) = 10$. *Solid line:* (4.8)*; dotted:* (4.1)*; dashed:* (3.8).

All the observations from the first example apply. Note that the two better techniques are nearly insensitive to increasing the condition number while the two other are getting worse. Also note that Figures 5.2 and 5.5 seem to suggest a general inferiority of INE using minimization only compared to original INE. This again supports the conjecture that INE is powerful when maximizing and weak when minimizing.

5.3. Example 3. We generated 50 matrices A of size 100 all with the same prescribed Euclidean norm N by choosing the uniformly distributed singular values

$$\sigma_k = \frac{N}{k}, \quad 1 \le k \le 100.$$

The matrix A was formed as $A = U\Sigma V^T$, where $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_{100})$, and the matrices U and V are the Q factors of the QR factorizations of matrices B generated using the MATLAB command `B=rand(100,100) - rand(100,100)`. Then we computed a column pivoting (using the MATLAB command `colamd(A)`) and obtained an upper triangular factor R from the QR decomposition of the column permuted matrix A. This

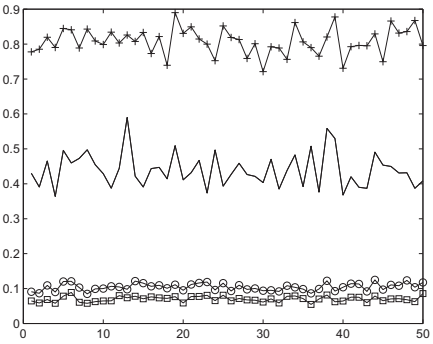


FIG. 5.5. *Ratio of estimate to real condition number for the* 50 *matrices in example* 2 *with* $\kappa(A) = 100$. *Solid line: ICE (original); pluses: INE with inverse and using only maximization; circles: INE (original); squares: INE with inverse and using only minimization.*

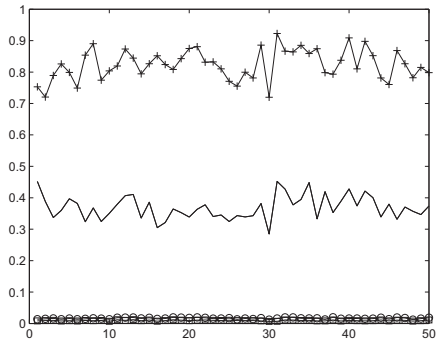


FIG. 5.6. *Ratio of estimate to real condition number for the* 50 *matrices in example* 2 *with* $\kappa(A) = 1000$. *Solid line: ICE (original); pluses: INE with inverse and using only maximization; circles: INE (original); squares: INE with inverse and using only minimization.*

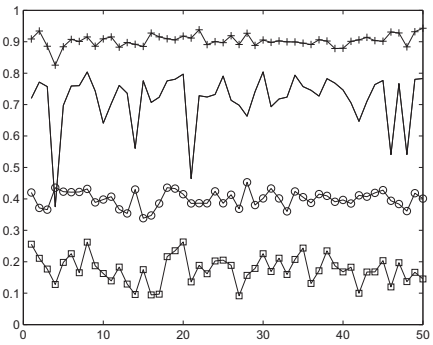


FIG. 5.7. *Ratio of estimate to real condition number for the* 50 *matrices in example* 3 *with* $N = 10$. *Solid line: ICE (original); pluses: INE with inverse and using only maximization; circles: INE (original); squares: INE with inverse and using only minimization.*

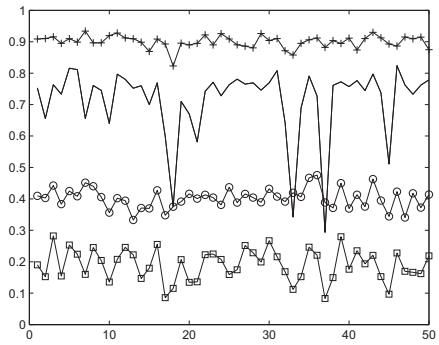


FIG. 5.8. *Ratio of estimate to real condition number for the* 50 *matrices in example* 3 *with* $N = 10^{12}$. *Solid line: ICE (original); pluses: INE with inverse and using only maximization; circles: INE (original); squares: INE with inverse and using only minimization.*

is the same type of experiment as tested in [18, section 5, Table 5.3]. The condition estimators were tested on R; see Figures 5.7 and 5.8 for, respectively, $N = 10, 10^{12}$. Qualitatively the same pictures are obtained when one omits column pivoting.

Again, INE with maximization only is the best for both cases of N. Also, the other techniques keep the same relative superiority as above (exception for one matrix in Figure 5.7 and two matrices in Figure 5.8). Further, all techniques perform better overall than with exponentially distributed singular values, even when the condition number is like that in Figure 5.5.

5.4. Example 4. We considered 20 small sparse matrices from the Matrix Market collection [17], most of them tested also in [18, section 5, Table 5.1]. We computed their QR decomposition (with and without column pivoting) and tested the estimators with the factor R. We provide the ratios of the ICE and INE estimates versus the actual condition numbers in Figures 5.9 and 5.10, with and without column pivoting

TABLE 5.1
Examples of matrices from Matrix Market: Ratios of the estimates over the actual condition numbers.

Number	Name	dim.	nnz	ICE (orig)	INE (orig)	INE (max)	INE (min)
1	494_bus	494	1666	0.09	0.06	0.99	0.02
1	(colamd)	494	1666	0.09	0.06	1	0.057
2	arc130	130	1037	0.42	4e-06	1	9e-10
2	(colamd)	130	1037	0.63	5e-06	1	5e-6
3	bfw398a	398	3678	0.29	0.005	0.83	0.004
3	(colamd)	398	3678	0.03	0.005	0.9	0.004
4	cavity04	317	5923	0.11	1e-4	0.88	3e-5
4	(colamd)	317	5923	0.13	5e-4	0.87	7e-6
5	ck400	400	2860	0.15	9e-5	0.99	8e-5
5	(colamd)	400	2860	0.09	2e-4	1	2e-5
6	dwa512	512	2480	0.16	0.005	0.97	0.003
6	(colamd)	512	2480	0.11	0.005	0.94	0.003
7	e05r0400	236	5846	0.09	5e-4	0.86	1e-4
7	(colamd)	236	5846	0.06	0.001	0.94	3e-4
8	fidap001	216	4339	0.63	0.02	0.76	0.01
8	(colamd)	216	4339	0.19	0.03	0.85	0.02
9	gre_343	343	1310	0.37	0.05	0.87	0.05
9	(colamd)	343	1310	0.33	0.025	0.9	0.023
10	impcol b	59	271	0.16	2e-4	0.98	5e-5
10	(colamd)	59	271	0.17	2e-4	0.98	5e-5
11	impcol c	137	400	0.24	0.007	0.99	0.007
11	(colamd)	137	400	0.32	0.006	0.99	0.006
12	lshp_406	406	2716	0.11	0.006	0.88	0.004
12	(colamd)	406	2716	0.13	0.006	0.88	0.005
13	lund_a	147	2449	0.18	3e-5	0.94	1e-5
13	(colamd)	147	2449	0.15	2e-4	0.91	1e-4
14	olm500	500	1996	0.08	0.03	0.93	0.019
14	(colamd)	500	1996	0.08	0.03	0.93	0.019
15	pde225	225	1065	0.38	0.11	0.77	0.088
15	(colamd)	225	1065	0.53	0.099	0.96	0.093
16	rw496	496	1859	0.92	3e-8	0.99	3e-8
16	(colamd)	496	1859	1e-5	3e-8	1	2e-8
17	saylr1	238	1128	0.4	0.07	0.69	0.02
17	(colamd)	238	1128	0.77	0.11	0.89	0.08
18	steam	240	2248	1	0.96	1	0.81
18	(colamd)	240	2248	1	0.2	1	0.03
19	str 0	363	2454	0.38	0.07	0.97	0.04
19	(colamd)	363	2454	0.06	0.08	0.71	0.02
20	west0381	381	2134	0.66	0.005	0.99	0.002
20	(colamd)	381	2134	0.4	0.003	0.92	0.002

by colamd, respectively. In these figures the x-axis corresponds to the matrix number, where the numbering follows from alphabetical ordering according to matrix name. In order to see the huge differences in the quality of the estimators we also provide the values of these ratios in Table 5.1. We can see that the differences between the individual techniques do change more among the matrices than in the previous examples, but the basic message is the same: the INE technique with maximization is the clear winner. Column pivoting seems to have a more profound influence. In some situations all techniques do reasonably well (the matrix "steam" without pivoting) or badly except for INE using only maximization (the matrix "rw496" with pivoting).

As above, we display for the matrices from the Matrix Market the fraction of cases in which the sufficient conditions for superiority of INE maximization (4.1), (4.8),

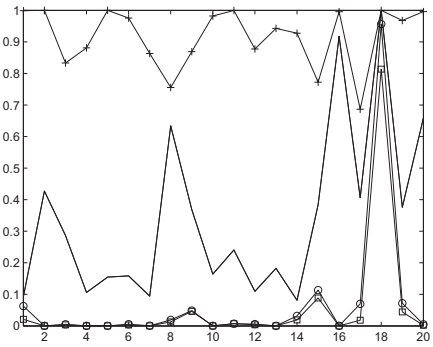


FIG. 5.9. *Ratio of estimate to actual condition number for the* 20 *matrices from the Matrix Market collection without column pivoting. Solid line: ICE (original); pluses: INE with inverse and using only maximization; circles: INE (original); squares: INE with inverse and using only minimization.*

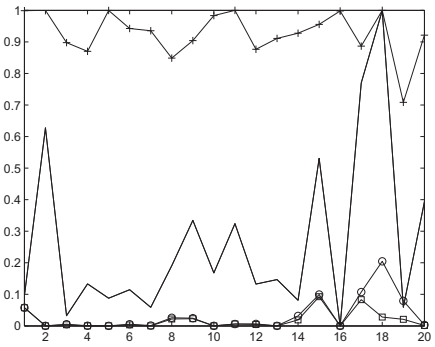


FIG. 5.10. *Ratio of estimate to actual condition number for the* 20 *matrices from the Matrix Market collection with column pivoting. Solid line: ICE (original); pluses: INE with inverse and using only maximization; circles: INE (original); squares: INE with inverse and using only minimization.*

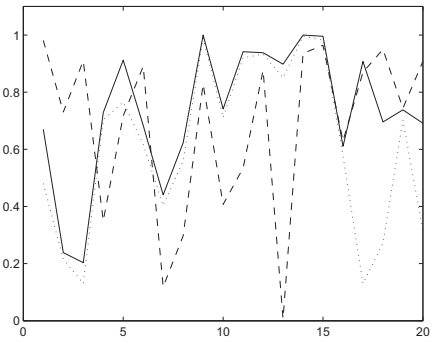


FIG. 5.11. *Ratio of the satisfied sufficient conditions in condition number estimation for the* 20 *matrices from the Matrix Market. Solid line:* (4.8)*; dotted:* (4.1)*; dashed:* (3.8).

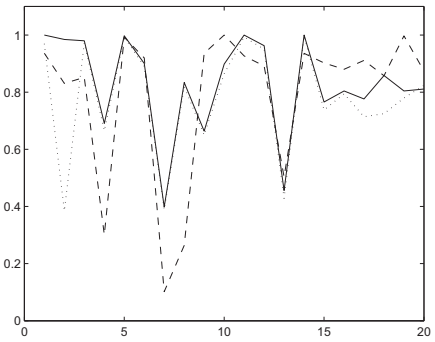


FIG. 5.12. *Ratio of the satisfied sufficient conditions in condition number estimation for the* 20 *matrices from the Matrix Market. Solid line:* (4.8)*; dotted:* (4.1)*; dashed:* (3.8).

and (3.8) are satisfied if this superiority is actually achieved. They are depicted in Figures 5.11 and 5.12. We can see that these conditions often seem to cover even more cases of INE maximization superiority than in the case of the random matrices from Example 1.

5.5. Example 5. The last series of experiments uses the investigated condition estimators inside a mixed direct-inverse matrix decomposition. As we mentioned in the introduction, we believe that more accurate estimates are also useful in an incomplete decomposition since their values may decide about dropping and pivoting. Here we use the compact BIF decomposition introduced in [12, 13] (see the MATLAB code there) that computes the incomplete direct and inverse factor at the same time and their mutual computation can be exploited in monitoring the decomposition. However, to facilitate comparison of the condition estimators, we will use only BIF decomposition without dropping, i.e., both the full direct and inverse factor are computed. Of course, in the case of the original ICE method we could use any

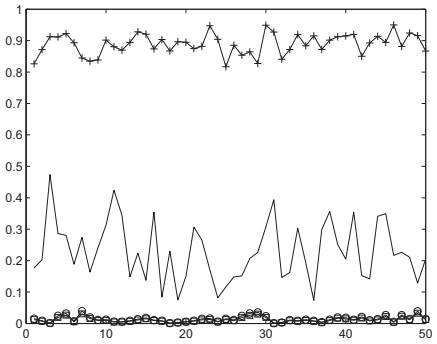


FIG. 5.13. *Ratio of estimate to actual condition number for the* 50 *dense symmetric positive definite matrices in example* 5 *decomposed with the BIF method. Solid line: ICE (original); pluses: INE with inverse and using only maximization; circles: INE (original); squares: INE with inverse and using only minimization.*

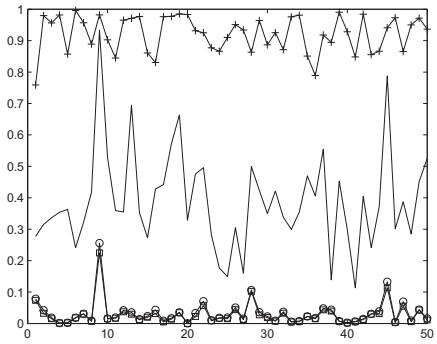


FIG. 5.14. *Ratio of estimate to actual condition number for the* 50 *sparse symmetric positive definite matrices in example* 5 *decomposed with the BIF method. Solid line: ICE (original); pluses: INE with inverse and using only maximization; circles: INE (original); squares: INE with inverse and using only minimization.*

other implementation of the Cholesky decomposition, but for simplicity we stick with the same method here.

First, we generated 50 dense symmetric positive definite matrices A of size 100 using the MATLAB command $\mathsf{B} = \mathsf{randn(100,100)}$ and putting $A = B^T B$. The results are displayed in Figure 5.13. Next we generated 50 sparse symmetric positive definite matrices A of size 100 using the MATLAB command $\mathsf{B} = \mathsf{sprandn(100,100,0.02)} + \mathsf{speye(100)}$ and putting $A = B^T B$. This gave matrices A with an average of about 850 nonzeros. The results are displayed in Figure 5.14.

As for Example 4, with sparse matrices the differences between the estimators are somehow less regular and sparse matrices seem to be favorable for original ICE. Nevertheless, the overall assessment of the quality of the individual techniques is as in the previous examples.

6. Conclusions and future work. In this paper, we have discussed incremental condition estimators in the 2-norm. In particular, the two main strategies, ICE and INE, were analyzed. It was shown that these two strategies are inherently different and the presented experiments support this claim. Moreover, we accumulated both theoretical and experimental evidence that the INE strategy using both the direct and the inverse factor is a method of choice yielding a highly accurate 2-norm estimator. Our future work will consider the effects of higher accuracy of the condition estimator used inside incomplete factorizations. In particular, we intend to use accurate condition estimation for dropping and pivoting. We also intend to develop a fast block version of the described strategy taking into account several ways to extract the estimates for the diagonal blocks.

Acknowledgments. We thank the two anonymous referees for their recommendations and Gérard Meurant for careful reading of the manuscript.

REFERENCES

[1] J. BENESTY AND T. GÄNSLER, *New insights into the RLS algorithm*, EURASIP J. Appl. Signal Processing, 3 (2004), pp. 331–339.

[2] J. Benesty and T. Gänsler, *A recursive estimation of the condition number in the RLS algorithm*, in Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing, vol. 4, 2005, pp. 25–28.
[3] C. H. Bischof, *Incremental condition estimation*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 312–322.
[4] C. H. Bischof, *A parallel QR factorization algorithm with controlled local pivoting.* SIAM J. Sci. Statist. Comput., 12 (1991), pp. 36–57.
[5] C. H. Bischof, J. G. Lewis, and D. J. Pierce, *Incremental condition estimation for sparse matrices*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 644–659.
[6] C. H. Bischof and P. T. P. Tang, *Generalizing incremental condition estimation*, J. Numer. Linear Algebra Appl., 1 (1992), pp. 149–163.
[7] M. Bollhöfer, *A robust ILU with pivoting based on monitoring the growth of the inverse factors*, Linear Algebra Appl., 338 (2001), pp. 201–218.
[8] M. Bollhöfer, *A robust and efficient ILU that incorporates the growth of the inverse triangular factors*, SIAM J. Sci. Comput., 25 (2003), pp. 86–103.
[9] M. Bollhöfer, *ILUPACK Version* 2.4, http://www.icm.tu-bs.de/ bolle/ilupack/ (2011).
[10] M. Bollhöfer and Y. Saad, *On the relations between ILUs and factored approximate inverses*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 219–237.
[11] M. Bollhöfer and Y. Saad, *Multilevel preconditioners constructed from inverse-based ILUs*, SIAM J. Sci. Comput., 27 (2006), pp. 1627–1650.
[12] R. Bru, J. Marín, J. Mas, and M. Tůma, *Balanced incomplete factorization.* SIAM J. Sci. Comput., 30 (2008), pp. 2302–2318.
[13] R. Bru, J. Marín, J. Mas, and M. Tůma, *Improved balanced incomplete factorization*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2431–2452.
[14] A. Cline, A. Conn, and C. Van Loan, *Generalizing the LINPACK condition estimator*, In Numerical Analysis, J. Hennart, ed., *Lecture Notes in Math.*, 909, Springer, Berlin, 1982, pp. 73–83.
[15] G. Cybenko, *Fast Toeplitz orthogonalization using inner products*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 734–740.
[16] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
[17] I. S. Duff, R. G. Grimes, and J. G. Lewis, *Sparse matrix test problems*, ACM Trans. Math. Softw., 15 (1989), pp. 1–14.
[18] I. S. Duff and C. Vömel, *Incremental norm estimation for dense and sparse matrices*, BIT, 42 (2002), 300–322.
[19] C. Fassino, *On updating the least singular value: A lower bound*, Calcolo, 40 (2003), pp. 213–229.
[20] R. Ferng, *Lanczos-Based Condition Estimation in Signal Processing and Optimization*, Ph.D. thesis, Department of Mathematics, North Carolina State University, 1991.
[21] W. R. Ferng, G. H. Golub, and R. J. Plemmons, *Adaptive Lanczos methods for recursive condition estimation*, Numer. Algorithms, 1 (1991), pp. 1–19.
[22] W. W. Hager, *Condition estimates*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 311–316.
[23] S. Haykin, *Adaptive Filter Theory,* 4th ed., Prentice-Hall, Englewood Cliffs, NJ, 2002.
[24] N. J. Higham, *A survey of condition number estimation for triangular matrices*, SIAM Rev., 29 (1987), 575–596.
[25] N. J. Higham, *FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation*, ACM Trans. Math. Softw., 14 (1989), pp. 381–396.
[26] N. J. Higham, *Experience with a matrix norm estimator*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 804–809.
[27] N. J. Higham and F. Tisseur, *A block algorithm for matrix* 1*-norm estimation, with an application to* 1*-norm pseudospectra*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1185–1201.
[28] J. Mandel and B. Sousedík, *Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods*, Comput. Methods Appl. Mech. Engrg., 196 (2007), pp. 1389–1399.
[29] N. Mastronardi, M. Van Barel, and R. Vandebril, *A Schur-based algorithm for computing bounds to the smallest eigenvalue of a symmetric positive definite Toeplitz matrix*, Linear Algebra Appl., 428 (2008), pp. 479–491.
[30] J. Neering, *Optimization and Estimation Techniques for Passive Acoustic Source Localization*, Ph.D. thesis, l'École nationale superieure des mines de Paris, 2009.
[31] C.-T. Pan and R. J. Plemmons, *Least squares modifications with inverse factorizations: Parallel implications*, J. Comput. Appl. Math., 27 (1989), pp. 109–127.

[32] D. J. Pierce and J. G. Lewis, *Sparse multifrontal rank revealing QR factorization*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 159–180.
[33] D. J. Pierce and R. J. Plemmons, *Fast adaptive condition estimation*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 274–291.
[34] D. J. Pierce and R. J. Plemmons, *Tracking the condition number for RLS in signal processing*, Math. Control Signals Systems, 5 (1992), pp. 23–39.
[35] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence, and Robustness*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
[36] G. M. Shroff and C. H. Bischof, *Adaptive condition estimation for rank-one updates of QR factorizations*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1264–1278.
[37] G. W. Stewart, *Matrix Algorithms. Volume* I*: Basic Decompositions*, SIAM, Philadelphia, 1998.
[38] A. C. N. van Duin, *Scalable parallel preconditioning with the sparse approximate inverse of triangular matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 987–1006.

SIAM J. SCI. COMPUT.
Vol. 29, No. 5, pp. 1918–1941

EFFICIENT PRECONDITIONING OF SEQUENCES OF NONSYMMETRIC LINEAR SYSTEMS*

JURJEN DUINTJER TEBBENS† AND MIROSLAV TŮMA†

Abstract. We present a new approach for approximate updates of factorized nonsymmetric preconditioners for solving sequences of linear algebraic systems. This approach is algebraic and it is theoretically motivated. It generalizes diagonal updates introduced by Benzi and Bertaccini [*BIT*, 43 (2003), pp. 231–244] and Bertaccini [*Electron. Trans. Numer. Anal.*, 18 (2004), pp. 49–64]. It is shown experimentally that this approach can be very beneficial. For example, it is successful in significantly decreasing the number of iterations of a preconditioned iterative method for solving subsequent systems of a sequence when compared with freezing the preconditioner from the first system of the sequence. In some cases, the updated preconditioners offer a rate of convergence similar to or even higher than the rate obtained when preconditioning with recomputed preconditioners. Since the updates are typically cheap and straightforward, their use is of practical interest. They can replace recomputing preconditioners, which is often expensive, especially in parallel and matrix-free environments.

Key words. preconditioned iterative methods, sparse matrices, sequences of linear algebraic systems, incomplete factorizations, factorization updates, Gauss–Jordan transformations, minimum spanning tree

AMS subject classifications. Primary, 65F10, 65F50, 65N22, 65H10; Secondary, 15A06

DOI. 10.1137/06066151X

1. Introduction.

We consider the solution of sequences of linear systems

$$A^{(i)}x = b^{(i)},\ i = 1, \ldots, \tag{1.1}$$

where $A^{(i)} \in \mathbb{R}^{n\times n}$ are general nonsingular sparse matrices and $b^{(i)} \in \mathbb{R}^n$ are corresponding right-hand sides. Such sequences arise in many applications such as computational fluid dynamics, structural mechanics, numerical optimization as well as in solving non-PDE problems. For example, a system of nonlinear equations $F(x) = 0$ for $F : \mathbb{R}^n \to \mathbb{R}^n$ solved by a Newton- or Broyden-type method leads to a sequence of problems

$$J(x_i)(x_{i+1} - x_i) = -F(x_i),\ i = 1, \ldots, \tag{1.2}$$

where $J(x_i)$ is the Jacobian evaluated in the current iteration x_i or its approximation [31], [32].

The solution of sequences of linear systems is the main bottleneck in many applications mentioned above. For instance, the solvers may need powerful preconditioners in order to be efficient, and computing preconditioners $M^{(1)}, M^{(2)}, \ldots$ for individual systems separately can be very expensive. There is a strong need for reduction of costs by sharing some of the computational effort among the subsequent linear systems.

A way to reduce the overall costs for solving systems of the type (1.2) is to modify Newton's method by skipping some Jacobian evaluations as in the Shamanskii

*Received by the editors May 31, 2006; accepted for publication (in revised form) January 8, 2007; published electronically September 28, 2007. This work was supported by the National Program of Research "Information Society" under project 1ET400300415.

http://www.siam.org/journals/sisc/29-5/66151.html

†Institute of Computer Science, Czech Academy of Sciences, Pod Vodárenskou věží 2, 18207 Praha 8, Czech Republic (tebbens@cs.cas.cz, tuma@cs.cas.cz). The work of the first author was supported by project KJB100300703 of the Grant Agency of the Academy of Sciences of the Czech Republic.

combination of Newton's method and the Newton-chord method [11], [54]. In this way we get a sequence of systems with identical matrices, and techniques for solving systems with more right-hand sides may be applied provided that the right-hand sides are available a priori; see, e.g., [45], [25], [55], [60]. However, combinations of Newton's method and the Newton-chord method have much weaker nonlinear convergence properties than the standard Newton method.

A different approach to reducing the overall costs, which is usually more efficient, is based on *freezing* the preconditioner (using the same preconditioner for a sequence of linear systems), but recomputing (approximate) Jacobians $A^{(i)}$ [12], [37], [38]. This approach is very natural in the context of a matrix-free environment, where the system matrices $A^{(i)}$ may be available only in the form of matrix-vector products (matvecs); see also the overview of matrix-free Newton–Krylov methods in [36].

Another way to avoid efficiency and/or memory related problems connected to algebraic preconditioning is to use conceptually simpler preconditioners derived from the physics of the problem. In some PDE problems the original operator can be replaced by a simpler one. Early results related to preconditioning by fast solvers can be found in [16], [24]. For instance, the simpler operator can be a scaled diffusion operator for a PDE with variable coefficients or a convection-diffusion operator [12], [34], [36]. In the algebraic setting, simple preconditioners derived from stationary iterative methods can be used. Preconditioning by the symmetric part of a nonsymmetric matrix was proposed in [17], [62]; see also [14]. Another popular preconditioning technique for general convection-diffusion-reaction models is based on generalizations of ADI splitting from [48]; see, e.g., [34]. Note that we restrict ourselves here to linear preconditioners; for nonlinear preconditioning techniques we refer, e.g., to [13] and the references therein. In order to make the preconditioning more efficient and to simplify the preconditioner setup even more, reformulations based on nested iterations were introduced; see, e.g., [59]. For instance, the flexible Krylov-subspace framework enables theoretically sound implementations of inner-outer Krylov-subspace methods [51], [56].

Freezing the preconditioner or using simple preconditioning techniques may not be enough for fast convergence in practice. Our contribution proposes new and efficient approximate updates of a preconditioner which is factorized as $LDU \approx A$. The updated preconditioners are then used for solving the subsequent members of the sequence. We do not assume any simple relation among the systems of the sequence. Note that straightforward approximate small rank preconditioner updates can be obtained in case of a sequence of linear systems from a quasi-Newton method, as shown in the symmetric and positive definite case in [44], [8]. It is well known how to compute the *exact* updates of sparse decompositions [19], [20], [21]; the techniques for *dense* updates starting in early papers, e.g., [27], and having mainly the intent of being applied to the simplex method of linear programming and its extensions are a classical part of numerical mathematics. Another algebraically motivated strategy used in preconditioning sequences of systems is to use adaptive information generated by Krylov-subspace methods [2]. Recent work on recycling explicit information from Krylov subspaces can be found in [41], [47].

In this paper we directly generalize the approximate diagonal updates which are useful for solving the parabolic PDEs proposed in [3]; see also [9]. This generalization consists in modifying general off-diagonal entries. Our numerical experiments show that the generalizations are competitive with recomputing the factorized nonsymmetric preconditioners in terms of achieving similar convergence rates for subsequent systems. Moreover, forming the updates can be significantly cheaper than recomput-

ing the preconditioner. As far as we know, there are no theoretical or experimental results in this direction. We give a couple of theoretical explanations for the good performance of the updates and discuss some unexpected effects which help to improve the convergence and, as far as we know, have not been communicated before. The strategy which we use forms the updated preconditioner from two separate layers: entries of the original factorized preconditioner and scaled entries of the matrix update. For the sake of quality and efficiency we typically need to exploit only a part of the update. This part may result from a Gauss–Seidel type of splitting, or it may be found in a more sophisticated way. In this paper we treat both cases.

The paper is organized as follows. In section 2 we present a brief introduction into preconditioner updates and motivate the basic form of our updated factorizations. In section 3 we describe the new techniques for approximate updating. The results of numerical experiments with the new algorithms are presented and discussed in section 4. Directions for current and future research are given in section 5. Throughout the paper, $\|\cdot\|$ denotes an arbitrary matrix norm.

2. The ideal updated preconditioner. Some of the strategies for updating preconditioners that we mentioned in the introduction are linked with specific classes of linear solvers (e.g., recycling Krylov subspaces) and nonlinear solvers (e.g., Broyden-type methods) or they were designed for symmetric matrices. In this paper we wish to consider sequences of general, nonsymmetric systems that are solved by preconditioned iterative methods. We address here the following problems: First, how can we update, in theory, a preconditioner in such a way that the updated preconditioner is likely to be as powerful as the original one? And second, how can we approximate, in practice, such an update in order to obtain a preconditioner that is inexpensive to apply and yet useful?

In order to simplify the notation, we consider two linear systems of dimension n denoted by $Ax = b$ and $A^+x^+ = b^+$. Denote the difference matrix $A - A^+$ by B and let M be a preconditioner approximating A. Some information about the quality of the preconditioner M can be taken from a norm of the matrix

$$A - M \tag{2.1}$$

or from some norm of the matrix

$$I - M^{-1}A \quad \text{or} \quad I - AM^{-1} \tag{2.2}$$

if we consider preconditioning from the left or right, respectively (see, e.g., [3]). If preconditioners are in factorized form, both (2.1) and (2.2) should be considered in practice since the preconditioners can suffer from two types of deteriorations. While the norm of the matrix (2.1) expresses *accuracy* of the preconditioner, the norms of the matrices (2.2) relate to its *stability* [15]; see also [5]. We will define updated preconditioners M^+ for A^+ whose accuracy and stability are close to the accuracy and stability of M for A. For their derivation we concentrate on the norm of the matrix (2.1) because of its simplicity. Later in this section we present theoretical results demonstrating that both accuracy and stability of the derived updates are comparable to or even better than those of M for A.

We immediately obtain

$$\|A - M\| = \|A^+ - (M - B)\|.$$

Hence $M^+ \equiv M - B$ represents an updated preconditioner for A^+ of the same "level" of accuracy as M represents for A. We will call it the *ideal* updated preconditioner.

Note that there may very well exist different preconditioners that are ideal with respect to a norm of $A^+ - M^+$. Just consider $M^+ = M - C$ for some matrix $C \neq B$ with

$$\|A - M\| = \|A^+ - M^+\| = \|A^+ - M + C\|.$$

Because B is often readily available, we will concentrate on $M^+ = M - B$.

If we want to use M^+ as a preconditioner, we need to multiply vectors with the inverse of M^+ in every iteration of the linear solver. In some problems, the difference matrix B is such that $(M - B)^{-1}$ can be obtained from M^{-1} with low costs. For instance, if B has small rank, M^+ can be easily inverted using the Sherman–Morrison formula; see, e.g., [44, 8]. In general, however, the ideal updated preconditioner cannot be used since multiplication of vectors with $(M - B)^{-1}$ is expensive. Instead, we will consider cheap approximations of $(M - B)^{-1}$.

In this paper we will assume that M is given in the form of a triangular decomposition as $M = LDU \approx A$, where L and U have unit main diagonal. The approximate updates of factorized preconditioners which we will describe below typically assume that the matrices have a strong diagonal. Note that this assumption is very similar to theoretical assumptions which are generally required to get simple incomplete factorizations without a breakdown. For example, standard ILU(0) and AINV preconditioners are proved to be breakdown-free if the system matrix is an H-matrix [43], [6]. In order to extend the breakdown-free property to more general matrices, we need to change the decomposition by modifications which make the diagonal stronger, e.g., by a preliminary shift [43], [40] (see also [33], [1]) or by global modification of the decomposition [57], [35], [4]. The transfer from diagonal dominance of the matrix to diagonal dominance of the factors is discussed, for example, in [7] (cf. [3]) or in the practical reordering strategies based on strong transversals [46], [22], [23]. In the following we tacitly assume matrices are given in such form that the factors L and U more or less approximate the identity matrix.

If $M - B$ is invertible, we can approximate its inverse by a product of more factors which are easier to invert. For example, we can replace $(M - B)^{-1}$ by a product of inverses of triangular matrices and by an inverse of a difference of matrices where a diagonal matrix is used instead of M, as in

$$(M - B)^{-1} = U^{-1}(D - L^{-1}BU^{-1})^{-1}L^{-1} \approx U^{-1}(D - B)^{-1}L^{-1}, \tag{2.3}$$

provided that $D - B$ is nonsingular. Now assume $\overline{D - B}$ is a nonsingular approximation of $D - B$ that can be inverted inexpensively. Then we can define a preconditioner M^+ via the last expression in (2.3) as

$$M^+ = L(\overline{D - B})U. \tag{2.4}$$

In the symmetric case, this preconditioner changes to $M^+ = L(\overline{D - B})L^T$; hence symmetry is preserved if we choose $\overline{D - B}$ appropriately. Here we are primarily interested in the nonsymmetric case, and in this case we can further simplify the update. For example, we can approximate as

$$(M - B)^{-1} = (DU - L^{-1}B)^{-1}L^{-1} \approx (DU - B)^{-1}L^{-1} \tag{2.5}$$

if $DU - B$ is nonsingular. If $\overline{DU - B}$ denotes a nonsingular and easily invertible approximation of $DU - B$, then we define M^+ by

$$M^+ = L(\overline{DU - B}). \tag{2.6}$$

In comparison with (2.4), it seems to be much easier to deal only with two factors. An analogue of (2.5) is approximation through

$$(M-B)^{-1} = U^{-1}(LD - BU^{-1})^{-1} \approx U^{-1}(LD-B)^{-1}. \tag{2.7}$$

In our experiments we choose between approximation with (2.5) or (2.7) adaptively (we explain this later on). We describe our theoretical results for the case (2.5) only.

A first question is whether the update (2.6) has the potential to be more powerful than the frozen preconditioner $M = LDU$ for A^+. In the following simple lemma we express the relation of the frozen preconditioner to the updated form quantitatively.

LEMMA 2.1. *Let* $\|A - LDU\| = \varepsilon\|A\| < \|B\|$. *Then the preconditioner from* (2.6) *satisfies*

$$\begin{aligned}\|A^+ - M^+\| &\le \frac{\|L(DU - \overline{DU-B}) - B\| + \varepsilon\|A\|}{\|B\| - \varepsilon\|A\|} \cdot \|A^+ - LDU\| \\ &\le \frac{\|L\|\,\|DU - B - \overline{DU-B}\| + \|L-I\|\,\|B\| + \varepsilon\|A\|}{\|B\| - \varepsilon\|A\|} \cdot \|A^+ - LDU\|.\end{aligned}$$

Proof. We get directly

$$\begin{aligned}\|A^+ - M^+\| &= \|A - B - L(\overline{DU-B})\| = \|(A - LDU) + L(DU - \overline{DU-B}) - B\| \\ &\le \big(\varepsilon\|A\| + \|L(DU - \overline{DU-B}) - B\|\big)\,\frac{\|B\| - \varepsilon\|A\|}{\|B\| - \varepsilon\|A\|} \\ &\le \big(\varepsilon\|A\| + \|L(DU - \overline{DU-B}) - B\|\big)\,\frac{\|(A - LDU) - B\|}{\|B\| - \varepsilon\|A\|} \\ &\le \|A^+ - LDU\|\,\frac{\|L(DU - \overline{DU-B}) - B\| + \varepsilon\|A\|}{\|B\| - \varepsilon\|A\|} \\ &= \|A^+ - LDU\|\,\frac{\|L(DU - \overline{DU-B} - B) + (L-I)B\| + \varepsilon\|A\|}{\|B\| - \varepsilon\|A\|} \\ &\le \|A^+ - LDU\|\,\frac{\|L\|\,\|DU - B - \overline{DU-B}\| + \|L-I\|\,\|B\| + \varepsilon\|A\|}{\|B\| - \varepsilon\|A\|}.\end{aligned}$$

∎

The multipliers of $\|A^+ - LDU\|$ in Lemma 2.1 can be smaller than one if $\overline{DU-B}$ is close to $DU - B$ and if $\|L - I\|$ tends to be small. In practice, taking into account preconditioner modifications to improve diagonal dominance, this is often realistic. Note that the assumption $\|A - LDU\| = \varepsilon\|A\| < \|B\|$ is satisfied as soon as we have a strong preconditioner $M = LDU$.

The lemma states, apart from showing a relation to the frozen preconditioner, that for $\varepsilon\|A\|$ small enough a good approximation to $\overline{DU-B}$ combined with a close to diagonal factor L yields an accurate preconditioner which *may* be as powerful as a recomputed preconditioner. If we have a recomputed preconditioner M^R with say, $\|A^+ - M^R\| = \delta = \|A - M\|$, then based on (2.5) we expect $\|A^+ - M^+\| \ge \delta$. But the previous lemma shows $\|A^+ - M^+\| < \delta$ is not at all excluded. In section 4 we will show experimentally that the update (2.6) in some cases gives a higher convergence rate than if the preconditioner is recomputed.

The following theorem shows in a different way that, under the given assumptions, the quality of the update may be better than that of recomputed preconditioners if the approximation $\overline{DU-B}$ is favorably chosen. Since Lemma 2.1 is related to the accuracy according to (2.1), the theorem considers its quality with respect to (2.2).

The result is a straightforward generalization of a result from [9]. To simplify the description, the scaled updated approximate factor $D^{-1}(\overline{DU-B})$ will be denoted by $\overline{U-D^{-1}B}$, and $U^{-1}(\overline{U-D^{-1}B})$ will be denoted by $I-\overline{U^{-1}D^{-1}B}$.

THEOREM 2.2. *Assume that $LDU+E=A$ for some error matrix E and let $\|\overline{U^{-1}D^{-1}B}\|_2 \le 1/c < 1$, where $\|\cdot\|_2$ denotes the Euclidean norm. Further assume that the singular values σ_i of*

$$(I-L)B + L\left(\overline{DU-B} - (DU+L^{-1}E-B)\right)$$

satisfy

$$\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_t \ge \delta \ge \sigma_{t+1} \ge \cdots \ge \sigma_n$$

for some integer t, $t \ll n$, and some small $\delta > 0$. Let $(\overline{DU-B})$ have nonzero main diagonal, and let $D = \operatorname{diag}(d_1,\ldots,d_n)$. Then there exist matrices F and Δ such that

$$(\overline{DU-B})^{-1}L^{-1}A^{+} = I + \Delta + F, \tag{2.8}$$

with rank$(\Delta) \le t$ and

$$\|F\|_2 \le \frac{c}{c-1}\max_i \frac{\delta}{|d_i|}\|L^{-1}\|_2\|U^{-1}\|_2.$$

Proof. We have

$$\begin{aligned} L(\overline{DU-B}) - A^{+} &= L(DU+L^{-1}E-B+\overline{DU-B}-(DU+L^{-1}E-B)) - A^{+} \\ &= (I-L)B + L\left(\overline{DU-B}-(DU+L^{-1}E-B)\right). \end{aligned}$$

By assumption, the SVD of the latter matrix can be written as

$$(I-L)B + L\left(\overline{DU-B}-(DU+L^{-1}E-B)\right) = W\Sigma V^T$$
$$= W\operatorname{diag}(\sigma_1,\ldots,\sigma_t,0,\ldots,0)V^T + W\operatorname{diag}(0,\ldots,0,\sigma_{t+1},\ldots,\sigma_n)V^T \equiv \Delta_1 + F_1,$$

where $\operatorname{rank}(\Delta_1) \le t$ and $\|F_1\|_2 \le \delta$. Hence

$$L(\overline{DU-B}) - A^{+} = \Delta_1 + F_1$$

and

$$(\overline{DU-B})^{-1}L^{-1}A^{+} = I - (\overline{DU-B})^{-1}L^{-1}\Delta_1 - (\overline{DU-B})^{-1}L^{-1}F_1.$$

By setting

$$F \equiv -(\overline{DU-B})^{-1}L^{-1}F_1, \qquad \Delta \equiv -(\overline{DU-B})^{-1}L^{-1}\Delta_1,$$

we get (2.8), where $\operatorname{rank}(\Delta) \le t$. The matrix F can be bounded by

$$\|F\|_2 \le \|L^{-1}\|_2 \left\| \left(D(\overline{U-D^{-1}B}) \right)^{-1} \right\|_2 \delta;$$

hence

$$\begin{aligned} \|F\|_2 &\le \max_i \frac{\delta}{|d_i|}\|L^{-1}\|_2\|(\overline{U-D^{-1}B})^{-1}\|_2 \\ &\le \max_i \frac{\delta}{|d_i|}\|L^{-1}\|_2\|U^{-1}\|_2\|(I-\overline{U^{-1}D^{-1}B})^{-1}\|_2. \end{aligned}$$

By assumption, $\|\overline{U^{-1}D^{-1}B}\|_2 \leq 1/c < 1$, and, consequently,

$$\begin{aligned}\|F\|_2 &\leq \max_i \frac{\delta}{|d_i|}\|L^{-1}\|_2\|U^{-1}\|_2 \left(1 - \|\overline{U^{-1}D^{-1}B}\|_2\right)^{-1} \\ &\leq \frac{c}{c-1}\max_i \frac{\delta}{|d_i|}\|L^{-1}\|_2\|U^{-1}\|_2. \qquad \square\end{aligned}$$

Note that if the matrix F in (2.8) is zero, then the preconditioned system is a rank t update of the identity, and Krylov-subspace methods converge, in exact arithmetics, in at most $t+1$ iterations.

In the following section we propose approximations $\overline{DU-B}$ of $DU-B$ that can be efficiently computed and that lead to preconditioners that are inexpensive to apply. All techniques we present can be analogously formulated for updates of the form $(\overline{LD-B})U$ corresponding to (2.7).

3. Approximate preconditioner updates. We propose the following strategies to approximate $DU-B$ by an easily invertible matrix $\overline{(DU-B)}$. A first obvious but effective strategy is to set $\overline{DU-B} \equiv triu(DU-B)$, where $triu$ denotes the possibly sparsified upper triangular part (including the main diagonal). This results in the preconditioner

$$M^+ = L(DU - triu(B)), \tag{3.1}$$

which can be obtained entirely for free. The additional cost for applying this preconditioner is one triangular sweep with the triangular part of B if we store B and U separately. We may also merge them; then the additional sweep can be virtually free if the sparsity patterns of $triu(B)$ and U are close enough. We will call the update constructed by considering entries only from one triangular part the *structured update*. A trivial structured sparsification is given by

$$\overline{DU-B} \equiv \mathrm{diag}(DU-B),$$

which is a straightforward application of an approach from [3] to nonsymmetric problems.

As we show in the experiments, the simple update (3.1) and its analogue

$$M^+ = (LD - tril(B))U \tag{3.2}$$

seem to be powerful in many problems. One expects them to be particularly suited when one triangular part of B clearly dominates the other. The typical situation of that kind arises when matrices come from upwind/downwind discretization schemes. Nevertheless, as they take into account only one triangular part of the difference matrix B, there may be applications where important information is lost, leading to weak convergence. In the following we present a technique to replace $DU-B$ by an easily invertible matrix which is in general not triangular.

Denote the matrices $\mathrm{diag}(\overline{DU-B})$ by $\tilde{D}$, and $\tilde{D}^{-1}(\tilde{D} - \overline{DU-B})$ by $\tilde{B}$, respectively. Then $\tilde{B}$ has zero diagonal and we can write

$$\overline{DU-B} = \tilde{D}(I-\tilde{B}). \tag{3.3}$$

To motivate the scaling transformation in (3.3) consider for a moment the case when $\tilde{B} = \beta e_i e_j^T$ for some $1 \leq i,j \leq n, i \neq j$, and recall that we assume $\overline{DU-B}$ is nonsingular; hence so is $I-\tilde{B}$. Then we get, with the Sherman–Morrison formula,

$$(I-\tilde{B})^{-1} = I + \beta e_i e_j^T/(1-\beta e_j^T e_i) = I + \beta e_i e_j^T = I + \tilde{B}. \tag{3.4}$$

The matrix in (3.4) is equal to the identity modified by an off-diagonal entry β at the position (i,j). That is, $(I-\tilde{B})$ is a special Gauss–Jordan transformation [28], it is inverted without costs, and it has a fill-in free inverse.

Based on this well-known fact, in the following we will try to find *unstructured* approximations $\overline{DU-B}$ of $DU-B$ such that the scaled matrix $I-\tilde{B}$ can be written as a product of Gauss–Jordan transformations

$$(I-e_{i_1}\tilde{b}_{i_1*})(I-e_{i_2}\tilde{b}_{i_2*})\dots(I-e_{i_K}\tilde{b}_{i_K*}), \qquad K\le n-1, \tag{3.5}$$

where $\tilde{B}=(\tilde{b})_{ij}$. Denote the sparsity structure of a row i of $\tilde{B}$ (with zero diagonal) by $row(i)$, that is, $row(i)=\{k|i\neq k\wedge\tilde{b}_{ik}\neq 0\}$. The multiplication $(I-\tilde{B})^{-1}v$ for a given vector v is very cheap, as stated in Observation 3.1.

OBSERVATION 3.1. *The number of operations for multiplying a vector by a matrix of the form* (3.5) *or its inverse is at most* $2\sum_{j=1}^{K}|row(i_j)|$.

It is well known that any unit upper triangular matrix $I-\tilde{B}$ from (3.3) can be trivially written as the product $R_{n-1}\dots R_1$ of $n-1$ elementary triangular matrices $R_i=I-e_i\tilde{b}_{i*}$ for $i=1,\dots,n-1$. Hence using (3.1) may be considered a special case of (3.5). The following theorem shows a necessary and sufficient condition for the existence of a decomposition of $I-\tilde{B}$ of the form (3.5).

THEOREM 3.1. *Let* $I-\tilde{B}=I-\sum_{j_l:l=1,\dots,K}e_{j_l}\tilde{b}_{j_l*}$. *Then*

$$I-\tilde{B}=(I-e_{i_1}\tilde{b}_{i_1*})(I-e_{i_2}\tilde{b}_{i_2*})\dots(I-e_{i_K}\tilde{b}_{i_K*}) \tag{3.6}$$

if and only if

$$i_l\notin\bigcup_{k=1}^{l-1}row(i_k)\ \text{for}\ 2\le l\le K \tag{3.7}$$

for all $i_1,\dots,i_K$ *such that* $\{j_1,\dots,j_K\}=\{i_1,\dots,i_K\}$.

Proof. The equivalence of (3.6) and (3.7) follows from the orthogonality of the unit vector e_{i_l} with respect to all $\tilde{b}_{i_k*}$ for $k<l$, $1\le l\le K$. ∎

Based on Theorem 3.1 we first propose a greedy procedure to find a suitable approximation $\overline{DU-B}$ with $I-\tilde{B}$ satisfying (3.6). Consider a sequential choice of indices $i_1,\dots,i_K$, where $K\le n-1$ are determined by the algorithm. In each step we keep and update a set of *candidate rows* $\mathcal{R}$ initialized by $\{1,\dots,n\}$. After choosing a row i we remove from $\mathcal{R}$ all the rows $j\in\mathcal{R}$ for which $\tilde{b}_{ij}\neq 0$.

ALGORITHM 3.1. We use this algorithm to approximate $DU-B$ by a matrix which, scaled by its diagonal, can be written in the form (3.6).

(1) *set* $\mathcal{R}=\{1,\dots,n\}$, $K=0$
(2) *for* $k=1,\dots,n$ *do*
(3) *set* $row(k)=\{i|i\neq k\wedge|(DU-B)_{ki}|\neq 0\}$
(4) *set* $p_k=\sum_{j\in row(k)}|(DU-B)_{kj}|$
(5) *end for*
(6) *while* $\mathcal{R}\neq 0$ *do*
(7) *choose a row* $i\in\mathcal{R}$ *maximizing* $p_i-\sum_{j\in\mathcal{R}\cap row(i)}p_j$
(8) *set* $K=K+1$, $i_K=i$, $\mathcal{R}=\mathcal{R}\backslash\{row(i_K)\cup i\}$
(9) *end while*

The row indices $i_1,\dots,i_K$ provided by Algorithm 3.1 then determine the approximation in (3.3) with $I-\tilde{B}$ equal to the product (3.5). The heuristic criterion in step (7) aims, on the one hand, to choose the row of $DU-B$ with the largest entries. On

the other hand, it stimulates the choice of a row which results, based on condition (3.7), in removal of candidate rows with small entries. To balance between the two heuristics one may want to introduce a weighting parameter ω and use

(7′) *choose a row* $i \in \mathcal{R}$ *maximizing* $p_i - \omega \cdot \sum_{j \in \mathcal{R} \cap row(i)} p_j$.

Clearly, the algorithm may find more factors of (3.6) if there are fewer nonzero entries in the searched rows. Therefore it may be reasonable to perform some dropping strategy on-the-fly when running the algorithm by substituting step (3) with

(3′) *set* $row(k) = \{i | i \neq k \wedge |(DU - B)_{ki}| > tol\}$

for a predefined drop tolerance *tol*. Apart from tolerance-based dropping, sparsification based on the given mask may enhance the effectiveness of our strategy. Note that sparsification not only helps in covering as many rows as possible by Gauss–Jordan transformations, but it also leads to less expensive matvecs with the inverse of (3.5).

A more elegant and systematic way to get an unstructured update based on Gauss–Jordan transformations can be described by the following bipartite graph model. Let us define the bipartite graph of $(DU - B)$ as $G(DU - B) = (R, C, E)$, where $R = \{1, \ldots, n\}$, $C = \{1', \ldots, n'\}$ and $E = \{(i, j') | (DU - B)_{ij} \neq 0\}$. Then we have the following result.

THEOREM 3.2. *Consider a spanning forest* $T = (V_T, E_T)$ *of* $G(DU - B)$ *such that* $\{(i, i') | 1 \leq i \leq n\} \subseteq E_T$. *Then the matrix* $\overline{DU - B} \in \mathbb{R}^{n \times n}$ *with the entries defined by*

$$\overline{(DU - B)}_{ij} = \begin{cases} (DU - B)_{ij} & \text{if } (i, j') \in E_T, \\ 0 & \text{otherwise,} \end{cases}$$

scaled by its diagonal entries as in (3.3)*, can be expressed as a product of the form* (3.5).

Proof. First consider the case when the spanning forest T is not connected. Components of T induce a block-diagonal splitting of $\overline{DU - B}$, and matrices corresponding to individual blocks can be mutually multiplied in any order without causing any fill-in. Consequently, we can assume without loss of generality that T is connected and that T is a spanning tree. In the following we will show how to form the sequence of Gauss–Jordan transformations from the left to the right.

Our assumption implies that T contains at most $n - 1$ edges (i, j') with $i \neq j$. There exists a free row vertex $i \in R$ in T which is in T incident only to the edge (i, i') such that there is an edge $(k, i') \in E_T$ for some k. Set $i_1 = i$. Then remove from T the vertices $i \in R$, $i' \in C$ and all edges incident to them. Clearly, the updated tree T contains a free row vertex again. By repeating the choice of free row vertices and updates T in this way we get the sequence $i_1, \ldots, i_{n-1}$. If we rewrite as $I - \tilde{B}$ the matrix $\overline{DU - B}$ scaled by its diagonal, we have $I - \tilde{B} = (I - e_{i_1} \tilde{b}_{i_1 *})(I - e_{i_2} \tilde{b}_{i_2 *}) \ldots (I - e_{i_{n-1}} \tilde{b}_{i_{n-1} *})$ which proves the theorem. ∎

Theorem 3.2 implies the following algorithmic strategy to find a matrix $\overline{DU - B}$ which would approximate $DU - B$ and could be expressed as a product of Gauss–Jordan transformations.

ALGORITHM 3.2. We use this algorithm to find $\overline{DU - B}$ such that (3.6) is satisfied based on a bipartite graph of $DU - B$.

(1) Find a spanning forest $T = (V_T, E_T)$ of $G(DU - B)$ of maximum weight with edge weights $w_{ij} = |(DU - B)_{ij}|$ for $(i, j') \in E_T$ such that $\{(i, i') | 1 \leq i \leq n\} \subseteq E_T$.
(2) Find the entries of $\tilde{B}$ (and corresponding entries of $\overline{DU - B}$) as well as a feasible ordering of Gauss–Jordan factors for $i_1, \ldots, i_{n-1}$ in (3.5) with Theorem 3.2.

(3) For each $k = 2, \ldots, n$ add to $\overline{DU - B}$ all entries $(DU - B)_{i_k l}$ of $DU - B$ such that $l \in \{i_1, \ldots, i_{k-1}\}$.

Note that in the last step of Algorithm 3.2 we possibly put into $\overline{DU - B}$ many more nonzero entries than the $2n - 1$ entries provided by the weighted spanning forest. This is possible because of Theorem 3.1. The complexity of the weighted minimum spanning forest (here we need, in fact, a weighted maximum forest) is $O(m \log m)$ for the Kruskal algorithm [39] and $O(n + m \log m)$ for the Prim algorithm [50], where m is the number of edges in the graph G. Note, in addition, that we start with the partial spanning tree with the set of edges $\{(i, i') | 1 \leq i \leq n\}$. While in some cases the algorithms may seem time consuming, this procedure can provide useful updates. As in Algorithm 3.1, we can sparsify $DU - B$ by discarding entries smaller than a certain drop tolerance tol, which reduces the value of m and therefore also the computational complexity.

From Lemma 2.1 it is clear that the quality of the approximation of $DU - B$ may play a decisive role in the power of the preconditioner $M^+ = L(\overline{DU - B})$. In practice, the way that the original incomplete decomposition is constructed (scaling L during the construction, pivoting) can strongly support the quality of $\overline{DU - B}$. In order to use the most powerful type of update, in our experiments we switch adaptively between (3.1) and (3.2) based on the weighting of both triangular parts of B and use an unstructured update based on Algorithm 3.1 or 3.2 if its weighting is the most important. More precisely, we compute sums of magnitudes of entries in the triangular parts of the matrices and simulate runs of Algorithms 3.1 and 3.2 to get the sum of magnitudes of entries covered by the unstructured update. We then use the strategy which corresponds to the maximum value among these sums.

It can and often does happen that, in spite of the fact that the updated preconditioner loses some information about the system matrix, it yields a better convergence rate than if the preconditioner would be recomputed from the scratch. There are several possible explanations for this phenomenon. First, note that we showed theoretically in Lemma 2.1 and Theorem 2.2 that our updated preconditioners have the potential to be stronger than recomputed factorizations. In practice, it frequently happens that by updating the preconditioner we relate it to a previous decomposition which is more diagonally dominant than a recomputed decomposition. A part of the stable triangular factors is inherited and the update may even stabilize less stable factors of the initial factorization. Note that a modified old decomposition might be useful in general, but, e.g., in the related strategy [43], the size of the modification should be typically rather small to get a useful preconditioned iterative method. This is exactly what happens when modifying with entries of difference matrices B that are typically small compared to those of $A^{(i)}$. In addition, updates appear to perform better also in cases where there is no instability. We presume this is so because the preconditioner may be favorably modified by the additional structural information given by the update. To our knowledge, this conjecture is stated for the first time. An overlooked fact is that the most powerful dual-threshold incomplete decompositions and inverse decompositions can be very memory-efficient, but they may discard the structure of the problem. Our updates can add to a memory-efficient decomposition cheap and useful information about the structure, as seems to be clear from our experiments. We believe that such a strategy might be used to improve constructing general preconditioners in some cases. We might consider the update as a simple and efficient way to modify off-diagonal entries of the preconditioner, thus getting a generalization of diagonal modifications from [43] or forced diagonal modifications introduced in [33]. It is not unusual that level-based incomplete decompositions are

much better than their sophisticated counterparts. Such a behavior has been observed on some VENKAT matrices from the Harwell–Boeing collection, where powerful and compact dual threshold ILUT [52] preconditioners are less efficient than often very dense but reasonably structured ILU preconditioners using the concept of levels [61], [29].

The next section is devoted to numerical experiments with the most promising updates introduced in the paper.

4. Numerical experiments. In this section we present results of numerical experiments with preconditioned Krylov-subspace methods for solving sequences of systems of linear algebraic equations, where updated preconditioners are compared with recomputed and frozen preconditioners. We consider the sequences in three application problems. The first and second problems were generated with the optimization software UFO [42]. The last application is based on [10]. Software for the problem was kindly provided by Philipp Birken. We present results with several kinds of ILU preconditioners to show that the introduced techniques are quite general. In order to show a larger spectrum of various results, some of the computations were done in MATLAB using its ILU decomposition script. We used MATLAB version 7.0. Most of the tests, in particular for larger problems, were written in Fortran 90 and were compiled by Compaq Visual Fortran 6.6a. The codes were run on a computer with Intel Pentium 4, 3GHz processor, 1GB RAM memory, and 512k L2 cache.

As an accelerator, the BiCGSTAB [58] iterative method with right preconditioning was used. We also performed some experiments with the restarted GMRES [53] method and the transpose-free QMR [26] method. The results were similar, and we do not report on them here. Iterations were stopped when the Euclidean norm of the residual was decreased by seven orders of magnitude. Nevertheless, in our experiments we observed close to linear behavior of convergence curves of the preconditioned iterative method. Therefore, we expect qualitatively the same results for weaker or nonuniform stopping criteria used in nonlinear solvers.

Our first test problem is a two-dimensional nonlinear convection-diffusion model problem which we use to illustrate various aspects of the proposed strategies (general behavior of the strategies, choice of parameters, values of the bounds in Lemma 2.1). It has the form (see, e.g., [31])

$$-\Delta u + Ru\left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}\right) = 2000x(1-x)y(1-y) \tag{4.1}$$

on the unit square, discretized by 5-point finite differences on a uniform 70×70 grid. The initial approximation is the discretization of $u_0(x, y) = 0$. We choose the modest Reynolds number $R = 50$ in order to avoid potential discretization problems which may ask for adding stabilization terms. We obtain a small sequence of 7 matrices with 24220 nonzeros each (in the tables we denote the number of nonzeros by nnz).

Our update techniques are particularly beneficial when recomputing preconditioners is expensive. We start with a typical example given by the so-called ILU(0) incomplete decomposition which has the same sparsity pattern as the matrix it preconditions. This has the obvious advantage that it enables straightforward a priori allocation, but its computation may be time-consuming. In Table 1 we display the total time to solve the whole sequence and the numbers of BiCGSTAB iterations needed to solve the individual linear systems for several preconditioning strategies. In the first, denoted by "Recomp," the ILU(0) preconditioner was computed for each

TABLE 1
Nonlinear convection-diffusion model problem with $R = 50$, $n = 4900$, $nnz = 24220$, $ILU(0)$.

ILU(0), psize $\approx$ 24000					
Matrix	Recomp	Freeze	Str.	Unstr. GJ	Unstr. Kr.
$A^{(0)}$	40	40	40	40	40
$A^{(1)}$	29	36	32	39	30
$A^{(2)}$	21	39	27	34	30
$A^{(3)}$	20	48	26	33	24
$A^{(4)}$	17	55	26	31	26
$A^{(5)}$	16	58	29	29	30
$A^{(6)}$	15	50	22	24	26
$A^{(7)}$	15	62	26	28	29
$A^{(8)}$	17	68	28	30	31
$A^{(9)}$	15	71	27	28	28
$A^{(10)}$	15	51	24	29	28
Overall time	11 s	7.5 s	5 s	8.5 s	12.5 s

matrix separately. The strategy "Freeze" used a fixed preconditioner. The strategy denoted by "Str" used structured updates, "Unstr. GJ" stands for unstructured updates based on Gauss–Jordan transformations obtained from Algorithm 3.1, and "Unstr. Kr." stands for those obtained from Algorithm 3.2, where the spanning tree is computed with the Kruskal algorithm. We see that the recomputed ILU(0) decompositions yield powerful preconditioners for our problem, but they are rather slowly computed in MATLAB. Freezing the initial ILU(0) decomposition avoids these slow computations, and although it yields much higher numbers of BiCGSTAB iterations, the overall time to solve the sequence is shorter. Excellent behavior of the structured updates is demonstrated by this table. Here the triangular parts were chosen adaptively based on the magnitudes of their entries. While iteration numbers are nearly as low as with recomputation, significant time savings are achieved by avoiding the recomputation of preconditioners. The iteration counts for unstructured updates from Algorithm 3.1 are a little higher than for structured updates, but they are clearly lower than with the frozen preconditioner. Unstructured updates from Algorithm 3.2 yield iteration numbers comparable to those of structured updates.

Of course, running Algorithm 3.1 or 3.2 to compute the unstructured updates adds a time penalty. However, the timings displayed in Table 1 are pessimistic because they include solving with nontriangular factors of the form (3.5), which cannot compete with the highly optimized implementation of backward and forward solves in MATLAB. The complexity of Algorithm 3.1 or 3.2 alone is not very high for sparse matrices since it is linear in the number of matrix nonzeros. In this context, note that using a drop tolerance in Algorithms 3.1 and 3.2 has an influence on the number of nonzeros and hence also on computational time. We computed the unstructured updates with $tol = 0.3$ in Algorithms 3.1 and 3.2. In practice this parameter should be chosen according to the following considerations for the individual algorithms.

In Algorithm 3.2 we first construct a maximum spanning forest of at most $2n-1$ entries. Hence we need a value of tol selecting the $2n-1$ largest entries and as few other entries as necessary to be able to build the spanning forest. We could have optimized the choice of tol according to this rationale, leading to $tol = 0.35$ and an overall time of 10.5 seconds. For Algorithm 3.1 the situation is quite different. Here, an interesting fact is that if we significantly overestimate the parameter, then the

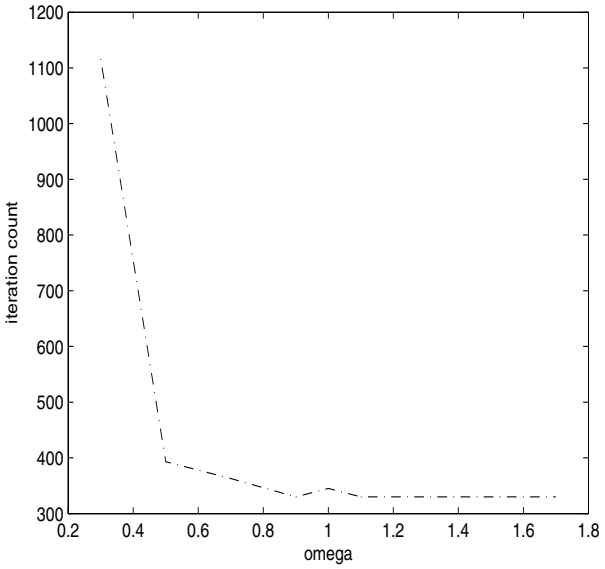


FIG. 1. *Nonlinear convection-diffusion model problem: Iteration counts for Unstr. GJ in dependency of* ω.

unstructured update may be very sparse since a smaller number of nonzeros can be covered by Gauss–Jordan transforms. If we underestimate it, then the update may be very sparse as well since we get only a small number of factors in the unstructured update of the form (3.1). In our case we did not optimize its choice, but a value $tol = 0.1-0.4$ for a reasonably scaled system matrix in order to keep only a few, say up to k, nonzeros in a row, and thus to cover by the unstructured update approximately $k \cdot n$ off-diagonal entries, is fine. This type of behavior is different from what we can sometimes observe in the field of algebraic preconditioners. As for the choice of ω in Algorithm 3.1, its value does not seem to have a crucial influence on the performance of the update either. In Figure 1 we display the total number of BiCGSTAB iterations needed to solve the whole sequence for different values of ω. If the values of ω are smaller than 0.5, criterion (7′) of Algorithm 3.1 starts to overemphasize the weight of the chosen candidate row, resulting in bad approximations of $DU - B$. In the other experiments presented here, we always used the choice $\omega = 1$.

In Table 2 the accuracies $\|A^{(i)} - M^+\|$ (in the Frobenius norm) of the preconditioners M^+ for the individual strategies are displayed. For this sequence, where stability of the preconditioners is not an issue, the accuracies correspond nicely to the numbers of BiCGSTAB iterations. We also present some information about the quality of the approximations $\overline{DU - B}$. Table 3 contains the values of the approximations $\|DU - B - \overline{DU - B}\|$ in the Frobenius norm for the considered update techniques. Also these values correspond to the numbers of BiCGSTAB iterations.

In Table 4 we take a closer look at the various update techniques we introduced. Whereas Table 1 suggests that structured updates provide more efficient preconditioners than unstructured updates, this is not apparent from Table 4. Here we use as initial preconditioner the ILU implemented in MATLAB with drop tolerance 0.01. The tol-

TABLE 2
Nonlinear convection-diffusion model problem, accuracies $\|A^{(i)} - M^+\|$.

ILU(0), psize ≈ 24000					
Matrix	Recomp	Freeze	Str.	Unstr. GJ	Unstr. Kr.
$A^{(0)}$	28.5	28.5	28.5	28.5	28.5
$A^{(1)}$	27.8	34.6	29.2	50.2	37.3
$A^{(2)}$	26.8	42.3	41.7	51.0	42.1
$A^{(3)}$	25.5	51.0	48.5	55.8	48.9
$A^{(4)}$	24.1	60.4	55.8	64.0	56.5
$A^{(5)}$	23.6	63.5	58.3	63.9	59.1
$A^{(6)}$	23.1	66.6	60.6	64.9	61.6
$A^{(7)}$	23.1	66.6	60.6	64.9	61.5
$A^{(8)}$	23.1	66.5	60.6	64.9	61.5
$A^{(9)}$	23.1	66.5	60.6	64.9	61.5
$A^{(10)}$	23.1	66.5	60.6	64.9	61.5

TABLE 3
Nonlinear convection-diffusion model problem, approximation qualities $\|DU - B - \overline{DU - B}\|$.

ILU(0), psize ≈ 632000			
Matrix	Str.	Unstr. GJ	Unstr. Kr.
$A^{(1)}$	13.89	37.01	18.19
$A^{(2)}$	22.1	36	22.7
$A^{(3)}$	29.78	40.1	30.34
$A^{(4)}$	37.46	48.32	38.47
$A^{(5)}$	39.92	47.39	41.2
$A^{(6)}$	42.29	47.91	43.69
$A^{(7)}$	42.27	47.9	43.66
$A^{(8)}$	42.23	47.86	43.62
$A^{(9)}$	42.23	47.86	43.63
$A^{(10)}$	42.23	47.86	43.63

TABLE 4
Nonlinear convection-diffusion model problem with $R = 50$, $n = 4900$, $nnz = 24220$, $ILU(0.01)$.

ILU(10^{-2}), psize ≈ 52000				
Matrix	Freeze	Unstr. GJ	Struct	Unstr. Kr.
$A^{(0)}$	17	17	17	17
$A^{(1)}$	34	57	21	48
$A^{(2)}$	49	43	24	36
$A^{(3)}$	77	39	34	33
$A^{(4)}$	102	36	54	29
$A^{(5)}$	140	37	69	28
$A^{(6)}$	142	30	76	25
$A^{(7)}$	154	35	77	28
$A^{(8)}$	144	36	91	33
$A^{(9)}$	152	35	91	29
$A^{(10)}$	123	31	90	28
Overall time	14.5 s	7.5 s	9 s	9 s

erance in Algorithms 3.1 and 3.2 for unstructured updates is 0.3. Clearly, unstructured updates are more powerful than structured updates with this kind of initial factorization. This is caused by the fact that the approximations $\overline{DU - B}$ in (2.6) cover more large entries when we use unstructured updates. In the following we quantify this property for a difference matrix B from the middle of the sequence, $B = A^{(0)} - A^{(4)}$. For other difference matrices from the sequence we would obtain similar numbers. With $B = A^{(0)} - A^{(4)}$, nonzero entries in $DU - B$ are quite evenly distributed over both triangular parts. We have $\|striu(DU - B)\| \approx 80$ and $\|stril(DU - B)\| \approx 38$ in the Frobenius norm. Here $stril(\cdot)$ and $striu(\cdot)$ denote the strict lower and upper triangular matrix part, respectively. Hence the upper triangular part is dominating, but important entries may be found in the lower part, too, and they are lost with structured updates. The unstructured updates take into account both triangular parts. This is reflected by the Frobenius norms $\|striu(\overline{DU - B})\| \approx 70$ and $\|stril(\overline{DU - B})\| \approx 16$ for the approximation $\overline{DU - B}$ from Algorithm 3.1. With Algorithm 3.2 we obtain $\|striu(\overline{DU - B})\| \approx 58$ and $\|stril(\overline{DU - B})\| \approx 32$. Note that Algorithm 3.2 yields more nonzeros, which is explained by step (3) of the algorithm. In this context, also note that the number of nonzeros of the initial factorization and structured updates is about 52000, whereas unstructured updates have smaller numbers of nonzeros, about 39000–46000, which makes application of the unstructured updated preconditioner less expensive. This is one of the reasons why the unstructured updates are competitive, even with respect to timing, with structured ones, in spite of the time penalty to run Algorithm 3.1 or 3.2. The other reason is, of course, lower BiCGSTAB iteration numbers.

In situations as in Table 1, recomputing preconditioners is outperformed by our updates because of the high expenses of recomputing. When, on the other hand, recomputation is straightforward, updates need not be more effective. An example is given in Table 5 with as initial preconditioner the dual-threshold ILUT$(0.1, 5)$ decomposition, implemented in Fortran 90. The number of nonzeros in the incomplete LU decomposition is about 38000 (slightly differing for different matrices). Here the time spent for recomputation is very small due to the simple discretization stencil, and by far the most time is spent while solving with BiCGSTAB. Still, concerning iteration counts, the (adaptively chosen) structured updates perform only slightly worse than recomputation. Note that there is a strong overlap between the location of the nonzeros in B and in the preconditioner, but as above, we did not merge the triangular parts of the updated preconditioner. Table 6 shows similar behavior for a much larger problem with ILUT$(0.1, 3)$ as initial decomposition. Here we discretized (4.1) on a 282×282 grid, the matrices having dimension 79524. While evaluating Tables 5 and 6, it is important to realize that the timings may provide here only partial information. In case of matrix-free implementation we typically need to estimate the matrices first using, for example, graph coloring techniques [18], [49]. Our matrices have five diagonals and this implies that they can be estimated by at most seven matvecs. Namely, the number of matvecs corresponds to the number of colors needed to color the undirected graph of $A^T A$, the so-called intersection graph. Computing some of the standard preconditioners both *directly and efficiently* based on matvecs is a state-of-the-art challenging problem and can be very time-consuming. When using updates in a matrix-free environment, only part of the difference matrix needs to be estimated. In our cases the needed part of the difference matrix was always available from at most three matvecs, because the intersection graph of the (possibly permuted) triangular part of the matrix could be colored by only three colors.

TABLE 5
Nonlinear convection-diffusion model problem with $R = 50$, $n = 4900$, $nnz = 24220$, $ILUT(0.1, 5)$.

ILUT(0.1, 5), timep ≈ 0.01, psize ≈ 38000			
Matrix	Recomp	Freeze	Struct. update
$A^{(0)}$	25	25	25
$A^{(1)}$	25	33	26
$A^{(2)}$	23	47	27
$A^{(3)}$	19	58	27
$A^{(4)}$	18	83	27
$A^{(5)}$	17	88	28
$A^{(6)}$	16	119	28
$A^{(7)}$	16	114	27
$A^{(8)}$	17	107	27
$A^{(9)}$	17	111	28
$A^{(10)}$	17	123	27
Overall time	0.20 s	0.78 s	0.25 s

TABLE 6
Nonlinear convection-diffusion model problem with $R = 50$, $n = 79524$, $nnz = 615997$, $ILUT(0.1, 3)$.

ILUT(0.1, 3), timep ≈ 0.05, psize ≈ 632000			
Matrix	Recomp	Freeze	Struct. update
$A^{(0)}$	82	82	82
$A^{(1)}$	86	85	82
$A^{(2)}$	73	97	82
$A^{(3)}$	72	91	76
$A^{(4)}$	66	97	73
$A^{(5)}$	68	113	77
$A^{(6)}$	71	140	75
$A^{(7)}$	68	139	70
$A^{(8)}$	70	137	76
$A^{(9)}$	69	136	83
$A^{(10)}$	65	217	72
Overall time	17.4 s	31.0 s	19.4 s

In addition to the experiments presented here we also performed some experiments where the nonlinear problems were discretized by upwind schemes, leading to triangular difference matrices. As one can guess from the pattern, the results for solving the linear problems were rather good, but we typically needed more nonlinear iterations. Consequently, discretization by central differences was preferable.

Our second test problem is a smaller but rather difficult problem of dimension 2500. It consists of the two-dimensional driven cavity problem of the form

$$\Delta\Delta u + R\left(\frac{\partial u}{\partial y}\frac{\partial \Delta u}{\partial x} - \frac{\partial u}{\partial x}\frac{\partial \Delta u}{\partial y}\right) = 0$$

on the unit square, discretized by 13-point finite differences on a shifted uniform grid with 50×50 inner nodes [30]. The boundary conditions are given by $u = 0$ on $\partial\Omega$ and $\partial u(0, y)/\partial x = 0$, $\partial u(1, y)/\partial x = 0$, $\partial u(x, 0)/\partial x = 0$, and $\partial u(x, 1)/\partial x = 1$. The initial approximation is the discretization of $u_0(x, y) = 0$.

For the same reason as before, we choose modest Reynolds numbers. Even with

TABLE 7
Driven cavity problem with $R = 50$, $n = 2500$, $nnz = 31504$, $ILU(0.01)$.

ILU(0.01), psize ≈ 47000					
Matrix	Recomp	Freeze	Str.	Unstr. GJ	Unstr. Kr.
$A^{(0)}$	93	93	93	93	93
$A^{(1)}$	269	93	88	337	81
$A^{(2)}$	> 500	> 500	156	324	58
$A^{(3)}$	> 500	164	179	265	60
$A^{(4)}$	> 500	288	298	206	74
$A^{(5)}$	> 500	> 500	144	184	71
$A^{(6)}$	> 500	> 500	132	190	70
Overall time	∞	∞	8 s	17 s	6.5 s

TABLE 8
Driven cavity problem with $R = 10$, $n = 2500$, $nnz = 31504$, $ILU(0.01)$.

ILU(0.01), psize ≈ 47000					
Matrix	Recomp	Freeze	Str.	Unstr. GJ	Unstr. Kr.
$A^{(0)}$	84	84	84	84	84
$A^{(1)}$	84	87	95	91	91
$A^{(2)}$	312	183	119	95	113
$A^{(3)}$	261	198	119	103	134
$A^{(4)}$	352	> 500	190	149	164
$A^{(5)}$	259	> 500	163	204	164
$A^{(6)}$	291	183	150	217	144
Overall time	12.5 s	∞	7 s	12 s	11 s

modest Reynolds numbers we obtain sequences of linear systems that are hard to solve for the BiCGSTAB accelerator. As system matrices have 31504 nonzeros, we needed a relatively dense initial ILU preconditioner with 47000 nonzeros and with drop tolerance 0.01 from MATLAB to be able to solve the linear systems at all. Sparser preconditioners caused BiCGSTAB to stagnate for the initial linear system. In Tables 7 and 8 we show experiments executed in MATLAB with the initial ILU(0.01) preconditioner for $R = 50$ and $R = 10$, respectively. As before, by "overall time" we mean the time needed to solve the whole sequence, including preconditioner computations. In the columns "Unstr." we display the performance of unstructured updates computed with Algorithm 3.1 ($tol = 0.05$ for $R = 50$ and $tol = 0.02$ for $R = 10$) and Algorithm 3.2 ($tol = 0.7$ for $R = 50$ and $tol = 0.02$ for $R = 10$).

This problem represents the case where recomputing should be avoided for stability reasons. For instance, with $R = 50$, the recomputation of the incomplete factorization failed for the last 5 linear systems (giving the MATLAB warning "Incomplete upper triangular factor had 1 zero diagonal replaced by local drop tolerance"). In order to quantify instability we computed estimates of the 2-norms of the inverses of the factors of the used factorizations. For the initial decomposition we have $\|U^{-1}\|_2 \approx 41$ and $\|(LD)^{-1}\|_2 \approx 264$, but these norms grow rapidly for subsequent recomputed factorizations. In the second column of Table 9 the norms for $(LD)^{-1}$ are displayed; norms for U^{-1} grow similarly. Clearly, forward and backward substitution have become unstable. In the columns corresponding to updated factorizations we estimated $\|(\overline{LD} - \overline{B})^{-1}\|_2$. We see that higher estimates correspond in the majority of cases to higher iteration numbers. In the frozen preconditioner strategy, however, instability

TABLE 9
Driven cavity problem with $R = 50$, estimated Euclidean norms of inverses of first factor.

ILU(0.01), psize $\approx$ 47000					
Matrix	Recomp	Freeze	Str.	Unstr. GJ	Unstr. Kr.
$A^{(0)}$	264	264	264	264	264
$A^{(1)}$	$2 \cdot 10^3$	264	203	1069	185
$A^{(2)}$	$9 \cdot 10^5$	264	227	99	101
$A^{(3)}$	$8 \cdot 10^4$	264	326	291	130
$A^{(4)}$	$3 \cdot 10^5$	264	327	290	131
$A^{(5)}$	$2 \cdot 10^5$	264	327	290	131
$A^{(6)}$	$4 \cdot 10^5$	264	327	290	131

is not the cause of stagnation. We guess the frozen preconditioner fails to provide the structural information contained in updated factorizations. The results for $R = 10$ reflect similar phenomena in a weaker form. Structured and unstructured updates from Algorithm 3.2 yield the best results. In the case $R = 50$ the optimal choice $tol = 0.7$ results in particularly good performance of Algorithm 3.2, with respect to both time and iteration count.

We conclude this section with an application which leads to very large sequences of linear systems. They arise from numerical computation of steady vertical air flow through a level tunnel at a low Mach number subject to the gravitational force. The domain is a two-dimensional longitudinal section of the tunnel with the pressure and density varying only in the horizontal direction such that the gravitational term is balanced out by the pressure gradient. Neumann boundary conditions and Lax–Friedrichs fluxes were used. The gravitation term and the Euler equations were separated by a first-order operator splitting. For the discretization, the implicit Euler method combined with the first-order finite volume discretization in space was used. In every time step, one Newton step is performed in the flow solver only. More details can be found in [10], in particular in section 6.2. Our results were very similar for more variations of the problem.

Table 10 contains the results for two sequences from the linear systems for the described problem with a relatively coarse discretization grid. We used the dual-threshold ILUT$(0.001, 5)$ preconditioner, where the parameters were chosen in order to have a preconditioner size (that is, number of nonzeros) close to the size of the original matrix and such that the total number of matvecs (two in each iteration) to solve the initial system is reasonably small.

Here we show results only for some linear systems from the beginning of the sequences (as given by the superscripts); the whole sequence has more than 1000 linear systems. Three preconditioning strategies were tested: recomputation, freezing and updating. Updates were always related to the first matrix of the sequence. In the first sequence of Table 10, the preconditioner that is being frozen or updated was computed for the matrix $A^{(0)}$, and in the second sequence it was taken from the 30th linear system. The update strategy was implemented as a black-box routine which decides which of the updates (unstructured update from Algorithm 3.1 or 3.2, structured update based on the upper triangular part of the difference matrix, or structured update based on the lower triangular part of the difference matrix) is used, based on the sum of magnitudes of strong matrix entries. The structured updates store the update separately, although merging with the decomposition could provide even better timings. The results are characterized by the number of iterations

TABLE 10
Air flow in a tunnel, $n = 4800$*,* $nnz = 138024$*,* $ILUT(0.001, 5)$.

ILUT(0.001, 5), timep ≈ 0.05, psize ≈ 135798						
Matrix	Recomp		Freeze		Update	
	Its	Time	Its	Time	Its	Time
$A^{(5)}$	29	0.57	19	0.33	19	0.34
$A^{(10)}$	30	0.55	17	0.27	17	0.27
$A^{(15)}$	33	0.64	21	0.39	19	0.34
$A^{(20)}$	32	0.64	19	0.34	17	0.31
$A^{(25)}$	33	0.56	20	0.33	19	0.33
$A^{(30)}$	34	0.66	24	0.44	21	0.34
$A^{(35)}$	33	0.66	23	0.42	19	0.36
$A^{(40)}$	39	0.72	31	0.52	24	0.39
$A^{(45)}$	44	0.78	33	0.55	27	0.45
$A^{(50)}$	40	0.75	39	0.63	24	0.44
$A^{(55)}$	40	0.74	47	0.78	25	0.42
$A^{(60)}$	47	0.85	80	1.41	31	0.56
$A^{(65)}$	47	0.80	107	1.64	27	0.42
$A^{(70)}$	38	0.75	72	1.28	28	0.51
$A^{(75)}$	114	2.03	230	4.06	105	1.96
$A^{(80)}$	63	1.19	87	1.51	80	1.42
$A^{(35)}$	33	0.66	36	0.63	35	0.67
$A^{(40)}$	39	0.72	37	0.64	35	0.59
$A^{(45)}$	44	0.78	42	0.67	35	0.59
$A^{(50)}$	40	0.75	43	0.67	29	0.45
$A^{(55)}$	40	0.74	57	0.95	31	0.53
$A^{(60)}$	47	0.85	84	1.37	33	0.54
$A^{(65)}$	47	0.80	102	1.55	34	0.52
$A^{(70)}$	38	0.75	87	1.47	34	0.58
$A^{(75)}$	114	2.03	163	2.65	147	2.45
$A^{(80)}$	63	1.19	81	1.38	93	1.64

of the BiCGSTAB method and by the timings of the preconditioned iterative method required to solve the individual linear systems, including the time required to compute the preconditioner. The average time to compute the preconditioner is denoted by *timep*, and its average number of nonzeros is denoted by *psize*. These last two characteristics differ slightly in individual computations of a sequence of problems. Note that preconditioning this problem was necessary; the unpreconditioned method worked rather poorly.

From Table 10 we can see once more that freezing the preconditioner may not be enough for getting efficiently preconditioned iterative methods for all the systems. Freezing with updating is typically better in terms of the number of matvecs. The additional solve with the update may add a time penalty, but its influence seems to be limited. Clearly, by changing the matrix more and more the gap between the efficiency of freezing and updating gets larger up to some point where, of course, also the update is not sufficient anymore. We included this point in our table, but in practice this would be the moment to recompute a factorization. As in the previous problem, it seems that the update is even more powerful than the recomputed preconditioners in the sense of giving the smallest number of iterations among all three preconditioning strategies. This must be mainly caused by the fact that recomputation becomes less

stable as the sequence proceeds, as can be seen from the iteration numbers around the 75th linear system. However, the role of additional structural information provided by updates should not be underestimated. In Table 12 we will consider a sequence without instability regions where updates are still more powerful than recomputed factorizations.

TABLE 11
Air flow in a tunnel, $n = 9600$*,* $nnz = 277224$*,* $ILUT(10^{-7}, 30)$.

ILUT(10^{-7}, 30), timep ≈ 0.1, psize ≈ 283751						
Matrix	Recomp		Freeze		Update	
	Its	Time	Its	Time	Its	Time
$A^{(0)}$	3	0.13	3	0.13	3	0.13
$A^{(5)}$	3	0.13	3	0.03	3	0.03
$A^{(10)}$	4	0.15	4	0.05	5	0.05
$A^{(15)}$	4	0.15	5	0.06	6	0.06
$A^{(20)}$	5	0.15	6	0.06	7	0.09
$A^{(30)}$	7	0.18	7	0.08	8	0.11
$A^{(40)}$	8	0.23	14	0.16	14	0.17
$A^{(45)}$	9	0.23	18	0.17	20	0.23
$A^{(46)}$	11	0.24	22	0.23	16	0.18
$A^{(47)}$	11	0.23	18	0.19	16	0.18
$A^{(48)}$	15	0.29	23	0.25	22	0.26
$A^{(49)}$	15	0.30	23	0.25	22	0.29
$A^{(50)}$	16	0.33	24	0.23	19	0.23
$A^{(51)}$	27	0.48	31	0.38	25	0.33
$A^{(52)}$	47	0.69	33	0.34	27	0.31
$A^{(53)}$	44	0.73	33	0.39	23	0.29
$A^{(54)}$	67	1.12	54	0.61	32	0.43
$A^{(55)}$	92	1.49	196	2.23	56	0.84
$A^{(56)}$	76	1.21	131	1.48	40	0.54
$A^{(57)}$	79	1.33	81	1.05	51	0.80
$A^{(58)}$	52	0.91	45	0.59	34	0.51
$A^{(59)}$	50	1.02	40	0.63	38	0.65
$A^{(60)}$	32	0.74	961	15.3	440	7.98

Table 11 presents qualitatively the same results for a larger matrix. As above, a powerful ILUT preconditioner was chosen in order to provide small iteration counts and to have the number of nonzeros of the preconditioner similar to the number of nonzeros of the original matrix. Note that for most of the more difficult problems, the time needed to solve the linear system is the best for our updates. While, as above, there is a similar behavior of the iteration counts we also show results for more matrices around the point where the original frozen preconditioner stops being useful. Note that for some matrices the updated preconditioner behaves *much* better than the other strategies.

In Table 12 we consider discretization leading to matrices of a dimension about 60000. Most of the remarks on the previous two tables can be made here too, though we note that there are no instability regions anymore. As before, updates achieve an acceleration compared to recomputing of up to 90%. The relation to the freezing strategy is the same as for the corresponding problems of smaller dimension. A noteworthy difference with smaller dimensions is that the ratio of the average time to recompute the preconditioner ("timep") the time to solve the systems is much larger.

Hence avoiding recomputation becomes more important with larger dimensions. To conclude, let us mention the problem of recomputing related to a different preconditioner. This large air flow problem with the standard AINV(0.1) preconditioner with a number of nonzeros close to the number of nonzeros in the first matrix of the sequence converges in 12 iterations on average, the time to compute the preconditioner is 1.67 s, and time for the BiCGSTAB iterations is 0.25 s! We may assume that the role of avoiding frequent recomputations will be significantly increased in this case, but we did not follow this line.

TABLE 12
Air flow in a tunnel, $n = 59392$*,* $nnz = 1127211$*,* $ILUT(10^{-8}, 8)$.

ILUT($10^{-8}, 8$), timep ≈ 0.45, psize ≈ 1307000–1490000						
Matrix	Recomp		Freeze		Update	
	Its	Time	Its	Time	Its	Time
$A^{(0)}$	24	1.25	24	1.25	24	1.25
$A^{(2)}$	21	1.13	27	0.95	23	0.88
$A^{(4)}$	22	1.15	27	0.90	23	0.89
$A^{(6)}$	21	1.15	27	0.90	23	0.90
$A^{(8)}$	21	1.14	26	0.93	23	0.89
$A^{(10)}$	22	1.15	26	0.91	23	0.91
$A^{(12)}$	24	1.23	27	0.97	23	0.88
$A^{(14)}$	23	1.20	27	1.01	23	0.90
$A^{(16)}$	24	1.23	27	0.95	22	0.89
$A^{(18)}$	24	1.27	27	0.92	22	0.89
$A^{(20)}$	25	1.23	28	0.90	21	0.83
$A^{(22)}$	25	1.24	28	0.92	22	0.86
$A^{(24)}$	26	1.29	28	0.98	22	0.84
$A^{(26)}$	29	1.60	28	1.00	22	0.85
$A^{(28)}$	30	1.43	29	0.95	22	0.84
$A^{(30)}$	28	1.37	28	0.97	23	0.89
$A^{(32)}$	31	1.53	33	1.06	22	0.81
$A^{(34)}$	28	1.42	28	0.95	23	0.89
$A^{(36)}$	31	1.51	30	1.02	22	0.91
$A^{(38)}$	30	1.51	29	1.01	23	0.95

5. Conclusions. In this paper we proposed a couple of algebraic procedures which may be useful for solving sequences of systems of linear equations. The numerical experiments show that our updated preconditioners can be rather successful in practice, and the updates can often replace recomputation of preconditioners. In many cases, one would like to make the overall number of operations smaller with simple updates, and our experiments confirm that this is possible. In particular, the preconditioner update seems to be more advantageous than the other approaches if one of the following cases applies: if preconditioner computation is not cheap, if its recomputation is unstable, or if the update is structurally dominant, that is, if it covers a significant part of the difference matrices from subsequent problems. Nevertheless, there can be also different, and sometimes very strong, reasons for avoiding preconditioner recomputations. In matrix-free and/or parallel environments, which are currently quite common, any recomputation of a preconditioner may be expensive. This is especially true for strong algebraic preconditioners which are used for solving difficult problems. We intentionally used structured updates based on only one trian-

gular part. Part of our motivation was that we concentrated on finding methods for problems where the nonsymmetry is apparent. In addition, we are interested in the structured update since we expect possible cheap estimation of sparsified triangular matrices. This may be important in a matrix-free environment. Note that our unstructured updates are very close to permuted (and sparsified) triangular updates. We intend to present fully matrix-free results in the near future. Another issue which we are currently investigating is combination of approximate factorizations with various Gauss–Seidel type preconditioners to define updates.

An interesting problem which we would like to consider in the future is to find first a nonsymmetric permutation that transforms the system matrices into a form more suitable for one particular structured or unstructured update. In particular, this permutation may make one triangular part of the matrices heavier (in the sense of the sum of magnitudes of its entries) than the other triangular part. This may have a connection to the combinatorial method in Algorithm 3.2 for finding an unstructured update. The use of a weighted spanning tree strongly brings to mind the popular strategy of matchings-based nonsymmetric permutations which has significantly improved algebraic preconditioning in recent years [23], [5].

Acknowledgments. The authors thank Philipp Birken and Ladislav Lukšan for providing the software for solving the nonlinear problems and for useful instructions on working with it. They thank Andreas Meister for initiating application of the proposed techniques to the tunnel problem.

REFERENCES

[1] M. A. Ajiz and A. Jennings, *A robust incomplete Choleski-conjugate gradient algorithm*, Internat. J. Numer. Methods Engrg., 20 (1984), pp. 949–966.
[2] J. Baglama, D. Calvetti, G. H. Golub, and L. Reichel, *Adaptively preconditioned GMRES algorithms*, SIAM J. Sci. Comput., 20 (1998), pp. 243–269.
[3] M. Benzi and D. Bertaccini, *Approximate inverse preconditioning for shifted linear systems*, BIT, 43 (2003), pp. 231–244.
[4] M. Benzi, J. K. Cullum, and M. Tůma, *Robust approximate inverse preconditioning for the conjugate gradient method*, SIAM J. Sci. Comput., 22 (2000), pp. 1318–1332.
[5] M. Benzi, J. C. Haws, and M. Tůma, *Preconditioning highly indefinite and nonsymmetric matrices*, SIAM J. Sci. Comput., 22 (2000), pp. 1333–1353.
[6] M. Benzi, C. D. Meyer, and M. Tůma, *A sparse approximate inverse preconditioner for the conjugate gradient method*, SIAM J. Sci. Comput., 17 (1996), pp. 1135–1149.
[7] M. Benzi and M. Tůma, *Orderings for factorized sparse approximate inverse preconditioners,* SIAM J. Sci. Comput., 21 (2000), pp. 1851–1868.
[8] L. Bergamaschi, R. Bru, A. Martínez, and M. Putti, *Quasi-Newton preconditioners for the inexact Newton method*, Electron. Trans. Numer. Anal., 23 (2006), pp. 76–87.
[9] D. Bertaccini, *Efficient preconditioning for sequences of parametric complex symmetric linear systems*, Electron. Trans. Numer. Anal., 18 (2004), pp. 49–64.
[10] P. Birken, *Numerical Simulation of Flows at Low Mach Numbers with Heat Sources*, Ph.D. thesis, University of Kassel, Kassel, Germany, 2005.
[11] R. P. Brent, *Some efficient algorithms for solving systems of nonlinear equations*, SIAM J. Numer. Anal., 10 (1973), pp. 327–344.
[12] P. N. Brown and Y. Saad, *Hybrid Krylov methods for nonlinear systems of equations*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 450–481.
[13] X.-C. Cai and D. E. Keyes, *Nonlinearly preconditioned inexact Newton algorithms*, SIAM J. Sci. Comput., 24 (2002), pp. 183–200.
[14] K. Chen, *Matrix Preconditioning Techniques and Applications*, Cambridge Monogr. Appl. Comput. Math. 19, Cambridge University Press, Cambridge, UK, 2005.
[15] E. Chow and Y. Saad, *Experimental study of ILU preconditioners for indefinite matrices*, J. Comput. Appl. Math., 86 (1997), pp. 387–414.
[16] P. Concus and G. H. Golub, *Use of fast direct methods for the efficient numerical solution of nonseparable elliptic equations*, SIAM J. Numer. Anal., 10 (1973), pp. 1103–1120.

[17] P. Concus and G. H. Golub, *A generalized conjugate gradient method for nonsymmetric systems of linear equations*, in Computing Methods in Applied Sciences and Engineering, Lecture Notes in Econom. Math. Systems 134, R. Glowinski and J. L. Lions, eds., Springer-Verlag, Berlin, New York, 1976, pp. 56–65.
[18] A. R. Curtis, M. J. D. Powell, and J. K. Reid, *On the estimation of sparse Jacobian matrices*, J. Inst. Math. Appl., 13 (1974), pp. 117–119.
[19] T. A. Davis and W. W. Hager, *Modifying a sparse Cholesky factorization*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 606–627.
[20] T. A. Davis and W. W. Hager, *Multiple-rank modifications of a sparse Cholesky factorization*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 997–1013.
[21] T. A. Davis and W. W. Hager, *Row modifications of a sparse Cholesky factorization*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 621–639.
[22] I. S. Duff and J. Koster, *The design and use of algorithms for permuting large entries to the diagonal of sparse matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 889–901.
[23] I. S. Duff and J. Koster, *On algorithms for permuting large entries to the diagonal of a sparse matrix*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 973–996.
[24] H. C. Elman and M. H. Schultz, *Preconditioning by fast direct methods for nonself-adjoint nonseparable elliptic equations*, SIAM J. Numer. Anal., 23 (1986), pp. 44–57.
[25] P. F. Fischer, *Projection techniques for iterative solution of $Ax = b$ with successive right-hand sides*, Comput. Methods Appl. Mech. Engrg., 163 (1998), pp. 193–204.
[26] R. W. Freund, *A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems*, SIAM J. Sci. Comput., 14 (1993), pp. 470–482.
[27] P. E. Gill, W. Murray, and M. A. Saunders, *Methods for computing and modifying the LDV factors of a matrix*, Math. Comp., 29 (1975), pp. 1051–1077.
[28] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, London, 1996.
[29] D. Hysom and A. Pothen, *A scalable parallel algorithm for incomplete factor preconditioning*, SIAM J. Sci. Comput., 22 (2001), pp. 2194–2215.
[30] I. E. Kaporin and O. Axelsson, *On a class of nonlinear equation solvers based on the residual norm reduction over a sequence of affine subspaces*, SIAM J. Sci. Comput., 16 (1995), pp. 228–249.
[31] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, Frontiers Appl. Math. 16, SIAM, Philadelphia, 1995.
[32] C. T. Kelley, *Solving Nonlinear Equations with Newton's Method*, Fundam. Algorithms 1, SIAM, Philadelphia, 2003.
[33] D. S. Kershaw, *The incomplete Cholesky-conjugate gradient method for the iterative solution of systems of linear equations*, J. Comput. Phys., 26 (1978), pp. 43–65.
[34] D. Keyes, *Terascale implicit methods for partial differential equations*, in Contemp. Math. 306, AMS, Providence, RI, 2001, pp. 29–84.
[35] S. A. Kharchenko, L. Yu. Kolotilina, A. A. Nikishin, and A. Yu. Yeremin, *A reliable AINV-type preconditioning method for constructing sparse approximate inverse preconditioners in factored form*, Numer. Linear Algebra Appl., 8 (2001), pp. 165–179.
[36] D. A. Knoll and D. E. Keyes, *Jacobian-free Newton-Krylov methods: A survey of approaches and applications*, J. Comput. Phys., 193 (2004), pp. 357–397.
[37] D. A. Knoll and P. R. McHugh, *Newton-Krylov methods applied to a system of convection-reaction-diffusion equations*, Comput. Phys. Comm., 88 (1995), pp. 141–160.
[38] D. A. Knoll, P. R. McHugh, and D. E. Keyes, *Newton-Krylov methods for low Mach number compressible combustion*, AIAA J., 34 (1996), pp. 961–967.
[39] J. B. Kruskal, *On the shortest spanning subtree of a graph and the traveling salesman problem*, Proc. Amer. Math. Soc., 2 (1956), pp. 48–50.
[40] I. Lee, P. Raghavan, and E. G. Ng, *Effective preconditioning through ordering interleaved with incomplete factorization*, SIAM J. Matrix Anal. Appl., 27 (2006), pp. 1069–1088.
[41] D. Loghin, D. Ruiz, and A. Touhami, *Adaptive preconditioners for nonlinear systems of equations*, J. Comput. Appl. Math., 189 (2006), pp. 326–374.
[42] L. Lukšan, M. Tůma, J. Vlček, N. Ramešová, M. Šiška, J. Hartman, and C. Matonoha, *UFO* 2006. *Interactive System for Universal Functional Optimization*, Technical report V-977, ICS AS CR, Prague, 2006.
[43] T. A. Manteuffel, *An incomplete factorization technique for positive definite linear systems*, Math. Comp., 34 (1980), pp. 473–497.
[44] J. L. Morales and J. Nocedal, *Automatic preconditioning by limited memory quasi-Newton updating*, SIAM J. Optim., 10 (2000), pp. 1079–1096.
[45] D. P. O'Leary, *The block conjugate gradient algorithm and related methods*, Linear Algebra Appl., 29 (1980), pp. 293–322.

[46] M. Olschowka and A. Neumaier, *A new pivoting strategy for Gaussian elimination*, Linear Algebra Appl., 240 (1996), pp. 131–151.
[47] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti, *Recycling Krylov subspaces for sequences of linear systems*, SIAM J. Sci. Comput., 28 (2006), pp. 1651–1674.
[48] D. W. Peaceman and H. H. Rachford, Jr., *The numerical solution of parabolic and elliptic differential equations*, SIAM J. Appl. Math., 3 (1955), pp. 28–41.
[49] A. Pothen, F. Manne, and A. H. Gebremedhin, *What color is your Jacobian? Graph coloring for computing derivatives*, SIAM Rev., 47 (2005), pp. 629–705.
[50] R. C. Prim, *Shortest connection networks and some generalizations*, Bell System Tech. J., 36 (1957), pp. 1389–1401.
[51] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.
[52] Y. Saad, *ILUT: A dual threshold incomplete ILU decomposition*, Numer. Linear Algebra Appl., 1 (1994), pp. 387–402.
[53] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[54] V. E. Shamanskii, *On a modification of the Newton method*, Ukrain. Mat. Ž., 19 (1967), pp. 133–138.
[55] V. Simoncini and E. Gallopoulos, *An iterative method for nonsymmeric systems with multiple right-hand sides*, SIAM J. Sci. Comput., 16 (1995), pp. 917–933.
[56] V. Simoncini and D. B. Szyld, *Flexible inner-outer Krylov subspace methods*, SIAM J. Numer. Anal., 40 (2003), pp. 2219–2239.
[57] M. Tismenetsky, *A new preconditioning technique for solving large sparse linear systems*, Linear Algebra Appl., 154/156 (1991), pp. 331–353.
[58] H. A. van der Vorst, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.
[59] P. Vassilevski, *Preconditioning nonsymmetric and indefinite finite element matrices*, Numer. Linear Algebra Appl., 1 (1992), pp. 59–76.
[60] B. Vital, *Etude de quelques méthodes de résolution de problèmes linéaires de grande taille sur multiprocesseur*, Ph.D. thesis, Université de Rennes I, Rennes, France, 1990.
[61] J. W. Watts III, *A conjugate gradient truncated direct method for the iterative solution of the reservoir simulation pressure equation*, Soc. Petr. Eng. J., 21 (1981), pp. 345–353.
[62] O. Widlund, *A Lanczos method for a class of nonsymmetric systems of linear equations*, SIAM J. Numer. Anal., 15 (1978), pp. 801–812.

Available online at www.sciencedirect.com
ScienceDirect
Applied Numerical Mathematics 58 (2008) 1628–1641
APPLIED NUMERICAL MATHEMATICS
www.elsevier.com/locate/apnum

Preconditioner updates applied to CFD model problems ☆

Philipp Birken [a,*], Jurjen Duintjer Tebbens [b], Andreas Meister [a], Miroslav Tůma [b]

[a] *University of Kassel, Department of Mathematics, Heinrich-Plett-Str. 40, 34132 Kassel, Germany*
[b] *Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 2, Prague, Czech Republic*

Available online 17 October 2007

Abstract

This paper deals with solving sequences of nonsymmetric linear systems with a block structure arising from compressible flow problems. The systems are solved by a preconditioned iterative method. We attempt to improve the overall solution process by sharing a part of the computational effort throughout the sequence. Our approach is fully algebraic and it is based on updating preconditioners by a block triangular update. A particular update is computed in a black-box fashion from the known preconditioner of some of the previous matrices, and from the difference of involved matrices. Results of our test compressible flow problems show, that the strategy speeds up the entire computation. The acceleration is particularly important in phases of instationary behavior where we saved about half of the computational time in the supersonic and moderate Mach number cases. In the low Mach number case the updated decompositions were similarly effective as the frozen preconditioners.

Keywords: Finite volume methods; Update preconditioning; Krylov subspace methods; Euler equations; Conservation laws

1. Introduction

Finite volume methods are standard discretization schemes for both stationary and instationary problems in aerodynamics. As the CFL condition puts a severe restriction on the time step of explicit methods, time integration is often done implicitly. Using Newton's method for the appearing nonlinear equation systems, the problem of solving a partial differential equation numerically is transformed into the problem of solving a sequence of linear equation systems. In general, up to 80% of the CPU time for a flow solver is spent solving the linear systems. Thus, the major bottleneck in numerical simulation is the solution of the sequence of linear systems and there is a continuous demand to improve upon the existing methods.

Popular methods used in solving the large and sparse linear systems involved here include multigrid methods and Krylov subspace methods. Multigrid methods use multiple discretization levels and combine several techniques

☆ The work of the first and third author is supported by the German Science Foundation as part of the Sonderforschungsbereich SFB/TR TRR 30 "Prozessintegrierte Herstellung funktional gradierter Strukturen auf der Grundlage thermo-mechanisch gekoppelter Phänomene", project C2. The work of the second and fourth author is supported by the Program Information Society under project 1ET400300415. The work of the second author is also supported by project number KJB100300703 of the Grant Agency of the Academy of Sciences of the Czech Republic.
* Corresponding author.
E-mail address: birken@mathematik.uni-kassel.de (P. Birken).

0168-9274/$30.00 © 2007 IMACS. Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.apnum.2007.10.001

on the different levels (see, e.g. [23]). For some important classes of problems they are asymptotically optimal, but they can also be sensitive to changes of the problem [10]. Krylov subspace methods are based on projecting the large linear system to subspaces of small dimension (see, e.g. [21]). The subspaces are generated through multiplication of vectors by the system matrix, thus enabling exploitation of sparsity. In favorable cases, dominant properties become apparent at an early stage of computation and a satisfactory approximation to the solution can be obtained in a relatively small number of iterations. In practice, one often combines multigrid methods with Krylov subspace methods by using a method of one class as a preconditioner for a method of the other class (see, e.g. [26]). We will consider here Krylov subspace methods, but the techniques we describe may also be applied to other solvers. For the nonnormal linear systems that we have to solve, basically two classes of Krylov subspace methods may be used. In the first class, whose main representative is the GMRES method [22], we find methods that reduce residual norms in every iteration, but that must be restarted for reasons of storage and computational costs. The second class contains methods like BiCGSTAB [24], working with short recurrences but without guarantee that the process does not start to oscillate or does not break down. Often more important than the choice of the specific Krylov subspace method used is the choice of the preconditioner for the linear systems. For our problems, incomplete factorizations lead to good results that are in many cases hard to improve.

In order to speed up the solution process of the linear systems arising in CFD problems, we will not search for new and even more sophisticated linear solvers or preconditioners in this paper. Instead, we will try to accelerate the existing methods by considering the whole sequence of linear systems and by trying to share some of the computational effort throughout the sequence. In stationary and instationary problems linear systems are often close during many subsequent iterations of the nonlinear process. A well-known way to exploit this is by skipping some evaluations of the Jacobian in Newton's method, changing only the right-hand sides. Unfortunately, this leads to weaker convergence of the nonlinear process. Concerning preconditioning, closeness of system matrices has been taken advantage of only in a rather naive way. Very often, a preconditioner is recomputed periodically with some heuristic choice of period, and at a certain point it may be completely frozen [16].

In recent years, a few attempts to *update* preconditioners for large sparse systems have been made in the numerical linear algebra community. The main idea is to derive efficient preconditioners from previous systems of the sequence in a cheap way, thus avoiding the expensive computation of a new preconditioner. For instance, in case of a sequence of linear systems from a quasi-Newton method, straightforward approximate small rank updates can be useful (this is shown in the SPD case in [18,5]). SPD matrices and updates of incomplete Cholesky preconditioners are considered in [17]. In [2,6] approximate diagonal and tridiagonal preconditioner updates were introduced for sequences of parametric complex symmetric linear systems. This technique was generalized to approximate (possibly permuted) triangular updates for nonsymmetric sequences in [9]. Finally, recycling of Krylov subspaces by using adaptive information generated during previous runs has been used to update both preconditioners and Krylov subspace iterations (see [20,13,19] and [1]). Note that from the mentioned techniques only the last two are designed for sequences of nonsymmetric linear systems.

In this paper we investigate the effect of updating preconditioners on the speed of the solution process for some model problems from CFD. These are chosen from a broad range of Mach numbers to represent different well-known types of problems. The model problems lead to nonsymmetric linear systems and we will update the corresponding preconditioners based on the technique proposed in [9]. To our knowledge, this kind of strategy is applied to the CFD model problems for the first time. We will describe how we adapted the original technique in order to be used for the model problems. Then we demonstrate that the technique is able to speed up the solution of the involved linear systems, with an acceleration being particularly significant in phases with important changes between subsequent system matrices. In the next section we address the governing equations and the discretization we used for the numerical solution process. In Section 3 we say some words about solving the linear systems in general and then concentrate on the update technique. Among others, we present some new theoretical results and a detailed overview of the modifications for block systems. In Section 4 we display and discuss the results of numerical experiments with the model problems. Unless otherwise stated, $\|\cdot\|$ denotes an arbitrary matrix norm.

2. Governing equations and finite volume discretization

2.1. The Euler equations

The equations governing our model problems are the 2D Euler equations. These consist of the conservation laws of mass, momentum and energy, closed by an equation of state. Given an open domain $D \subset \mathbb{R}^2$, the equations can be expressed as

$$\partial_t \mathbf{u} + \sum_{j=1}^{2} \partial_{x_j} \mathbf{f}_j(\mathbf{u}) = \mathbf{0} \quad \text{in } D \times \mathbb{R}^+,$$

where $\mathbf{u} = (\rho, m_1, m_2, \rho E)^T$ represents the vector of conserved variables. The flux functions $\mathbf{f}_j$ are given by

$$\mathbf{f}_j(\mathbf{u}) = \begin{pmatrix} m_j \\ m_j v_1 + \delta_{1j} p \\ m_j v_2 + \delta_{2j} p \\ H m_j \end{pmatrix}, \quad j = 1, 2,$$

with δ_{ij} denoting the Kronecker symbol. The quantities ρ, $\mathbf{v} = (v_1, v_2)^T$, $\mathbf{m} = (m_1, m_2)^T$, E and $H = E + \frac{p}{\rho}$ describe the density, velocity, momentum per unit volume, total energy per unit mass and total enthalpy per unit mass, respectively. The pressure is defined by the equation of state for a perfect gas $p = (\gamma - 1)\rho(E - \frac{1}{2}|\mathbf{v}|^2)$, where γ denotes the ratio of specific heats, taken as 1.4 for air.

2.2. The finite volume method

We will use here a finite volume discretization. As this approach is covered extensively in the literature [12,15] we will give only a short summary of the specific concepts used. Our spatial discretization of the time independent physical domain into control volumes or cells σ_i is constructed as a secondary mesh from an underlying Delaunay-triangularization, see Fig. 1 (left). For a control volume σ_i with volume $|\sigma_i|$, let $N(i)$ denote the set of its neighbors. Then integration of the Euler equations over σ_i and the divergence theorem results in (see Fig. 1 (right) for the notation)

$$\frac{d}{dt}\mathbf{u}_i(t) = -\frac{1}{|\sigma_i|} \sum_{j \in N(i)} \sum_{k=1}^{2} \int_{l_{ij}^k} \sum_{\ell=1}^{2} \mathbf{f}_\ell(\mathbf{u}) \mathbf{n}_{ij,\ell}^k \, ds. \tag{1}$$

We now consider the mean value $\mathbf{u}_i(t) := \frac{1}{|\sigma_i|} \int_{\sigma_i} \mathbf{u} \, dx$ in each cell. The line integrals are computed using a second order Gaussian quadrature rule with Gauss point x_{ij}^k and a numerical flux function $\mathbf{H}$, which we have chosen to be

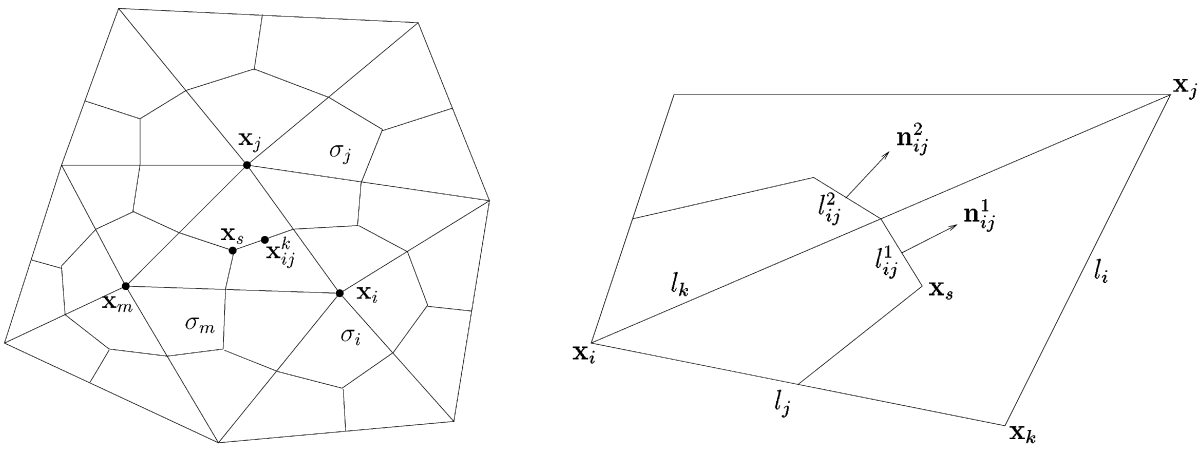


Fig. 1. Triangularization and boxes (left). Geometry between boxes (right).

AUSMDV from [25] or for low Mach numbers a Lax–Friedrichs-type flux developed for these cases [14]. Then, we obtain the following evolution equation for the cell averages on σ_i:

$$\frac{d}{dt}\mathbf{u}_i(t) = -\frac{1}{|\sigma_i|}\sum_{j\in N(i)}\sum_{k=1}^{2}|l_{ij}^k|\mathbf{H}\big(\mathbf{u}_i(t),\mathbf{u}_j(t);\mathbf{n}_{ij}^k\big). \tag{2}$$

To obtain higher order, we use a linear reconstruction technique, combined with the Barth–Jespersen-limiter to reduce the order where necessary.

Implicit time stepping schemes inherently fulfill the CFL stability condition, since the numerical domain of dependence always covers the physical one. In the numerical experiments we will consider the computation of steady states via timestepping with large time steps. Therefore, we employ the implicit Euler scheme and obtain the nonlinear system

$$\mathbf{\Omega u}^{n+1} = \mathbf{\Omega u}^n + \Delta t\mathbf{H}(\mathbf{u}^{n+1}),$$

where $\mathbf{u}$ is the vector of the conservative variables from all cells. Correspondingly, $\mathbf{H}(\mathbf{u})$ denotes an evaluation of the numerical flux function on the whole grid. $\mathbf{\Omega}$ is the diagonal matrix of the volumes of the cells, corresponding to the variables in $\mathbf{u}$. This equation is solved approximately using one step of Newton's method, which is sufficient for steady state problems. For unsteady problems more steps are often required and the extension of the method is straightforward. The starting value here is $\mathbf{u}^n$ and the corresponding linear system of equations can be written as (see (2))

$$\mathbf{A}\Delta\mathbf{u} = \mathbf{rhs}(\mathbf{u}^n), \quad \text{where } \mathbf{A} = \left[\mathbf{\Omega} + \Delta t\frac{\partial\mathbf{H}(\mathbf{u})}{\partial\mathbf{u}}\right]_{\mathbf{u}^n}, \tag{3}$$

with the update $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta\mathbf{u}$. The matrix $\mathbf{A} = (\mathbf{A}_{ij})$ has a block structure, where each element $\mathbf{A}_{ij} \in \mathbb{R}^{4\times4}$ vanishes if the corresponding control volumes σ_i and σ_j are not adjacent. Clearly, $\mathbf{A}$ represents a large and sparse matrix. As the involved grid is in general unstructured, so is the sparsity pattern of $\mathbf{A}$. Note that the sparsity pattern of these matrices remains the same during all time steps. Whereas in some cases at least the pattern is symmetric, usually the matrix itself is nonsymmetric. From (3) we can deduce that the matrix is close to a block diagonal matrix for small time steps and small derivatives of $\mathbf{H}(u)$. Diagonal dominance implies some attractive properties of preconditioners and iterative solvers; however, in our problems the dominance is too weak to take advantage of.

3. Iterative solution of the involved systems

3.1. Preconditioned Krylov subspace methods

As we mentioned in the introduction, we will solve the linear systems from (3) with Krylov subspace methods. For simplicity of notation, we denote linear systems from (3) by $\mathbf{A}x = b$. For the nonsymmetric matrices we have here, the choices of robust Krylov subspace methods are somewhat limited. A popular and efficient method with low demands on storage costs is the BiCGSTAB method [24]. Whereas the similarly popular GMRES method [22] has some other advantages that we explained in the introduction, we concentrate here on BiCGSTAB because for our finite volume scheme it has turned out to be slightly faster than GMRES. Of major importance for the performance of Krylov subspace methods is the choice of the preconditioner. From experience, right preconditioning seems to be the better choice in the context of compressible flows. Therefore, from now on we assume $\mathbf{M}$ is a right preconditioner approximating $\mathbf{A}$ which is applied as

$$\mathbf{A}M^{-1}\mathbf{x}^P = \mathbf{b}, \quad \mathbf{x} = \mathbf{M}^{-1}\mathbf{x}^P.$$

An overview of preconditioners with special emphasis on application in flow problems can be found in [16] and [7]. In our context, the most appropriate class of preconditioners is that of incomplete LU (ILU) decompositions. Here we focus on ILU(0), which has no additional level of fill beyond the sparsity pattern of the original matrix $\mathbf{A}$. This has the obvious advantage that it enables straightforward a priori allocation, and its memory demands are more predictable than for some other incomplete decompositions. Though ILU(0) may not be powerful enough for some difficult problems, for an important number of applications from CFD, including our model problems, it is efficient. In fact, as most

problems have a block structure, the used preconditioner is a *block* ILU(0) decomposition (BILU(0)) where pointwise operations are replaced by blockwise operations in the Gaussian elimination process. In our model problems, the blocks correspond to the 4×4 units the Jacobian consists of (see (3)). For the involved BILU(0) decompositions we use the following notation. We assume they are computed rowwise, hence the result is a block lower triangular factor denoted by $\mathbf{L}$ with 4×4 identity matrices on the main diagonal and a block upper triangular factor $\mathbf{U_D}$ with arbitrary nonsingular 4×4 matrices on the main diagonal. In addition, we denote by $\mathbf{D}$ the block diagonal part of $\mathbf{U_D}$ and let $\mathbf{U}$ be the matrix $\mathbf{U}_D$ scaled by $\mathbf{D}^{-1}$, i.e. $\mathbf{U} = \mathbf{D}^{-1}\mathbf{U_D}$. Then $\mathbf{U}$ has, like $\mathbf{L}$, 4×4 identity matrices on its main diagonal.

The main focus of this paper is efficient preconditioning of the *sequences* of linear systems arising from the scheme described above. Some strategies to share part of the computational effort throughout a sequence were mentioned in the introduction. The two tools we will use here are periodic *recomputation* of preconditioners combined with *approximate updating*. The idea of periodic recomputation is clear: Computing the preconditioner for every new linear system is time-consuming and unnecessary when the system matrices change slowly. Therefore, we will freeze preconditioners while solving several subsequent systems. Here we will not consider the problem of finding optimal recomputation periods or sophisticated strategies to adapt periods dynamically. This decision is supported by a set of experiments in which we failed to improve a fixed period for recomputation of the frozen preconditioner by simple adaptation guided by a reference number of iterations. The reason was that by simple adaptation to the iteration counts of our preconditioned iterative method we may fail to distinguish what are small/large numbers of iterations with respect to different phases of the problem. Different phases, which may be induced not only by the physics, but also by other adaptive procedures (e.g. for timestepping) may have completely different convergence properties. Therefore, dynamic strategies for preconditioner recomputations should be rather sophisticated. Instead, we will use periodically recomputed frozen preconditioners, which we found to perform rather well.

Our contribution concentrates on a way to update the frozen preconditioners to enhance their power. We believe that our strategy is easy to implement, parameter-free and with a small overhead. The technique we base our updates on is described in [9]. In the next section we have reformulated this strategy for the type of decomposition used here. We present several theoretical statements on the efficiency of the updates for the BILU(0) preconditioning. Furthermore, we give a detailed description of some implementation aspects which are relevant when applying the updates to our applications.

3.2. Preconditioner updates

In addition to a system $\mathbf{A}x = b$ with preconditioner $\mathbf{M} = \mathbf{LU_D} = \mathbf{LDU}$, let $\mathbf{A}^+x^+ = b^+$ be a system of the same dimension arising later in the sequence and denote the difference matrix $\mathbf{A} - \mathbf{A}^+$ by $\mathbf{B}$. We search for an updated preconditioner $\mathbf{M}^+$ for $\mathbf{A}^+x^+ = b^+$. We have

$$\|\mathbf{A} - \mathbf{M}\| = \left\|\mathbf{A}^+ - (\mathbf{M} - \mathbf{B})\right\|,$$

hence the level of accuracy of $\mathbf{M}^+ \equiv \mathbf{M} - \mathbf{B}$ for $\mathbf{A}^+$ is the same, in the chosen norm, as that of $\mathbf{M}$ for $\mathbf{A}$. The update techniques from [9] are based on the *ideal* updated preconditioner $\mathbf{M}^+ = \mathbf{M} - \mathbf{B}$. If we would use it as a preconditioner, we would need to solve systems with $\mathbf{M} - B$ as system matrix in every iteration of the linear solver. Clearly, for general difference matrices $\mathbf{B}$ the ideal updated preconditioner cannot be used in practice since the systems would be too hard to solve. We will consider cheap approximations of $\mathbf{M} - \mathbf{B}$ instead.

If $\mathbf{M} - \mathbf{B}$ is nonsingular, we approximate its inverse by a product of factors which are easier to invert. The approximation consists of two steps. First, we approximate $\mathbf{M} - \mathbf{B}$ as

$$\mathbf{M} - \mathbf{B} = \mathbf{L}(\mathbf{U_D} - \mathbf{L}^{-1}\mathbf{B}) \approx \mathbf{L}(\mathbf{U_D} - \mathbf{B}), \tag{4}$$

or by

$$\mathbf{M} - \mathbf{B} = (\mathbf{LD} - \mathbf{BU}^{-1})\mathbf{U} \approx (\mathbf{LD} - \mathbf{B})\mathbf{U}. \tag{5}$$

Next we replace $\mathbf{U_D} - \mathbf{B}$ or $\mathbf{LD} - \mathbf{B}$ by a nonsingular and easily invertible approximation. In [9] several options are proposed. We have here modified the first option in order to apply it to BILU(0) preconditioners and will approximate as

$$\mathbf{U_{D-B}} \approx \mathit{btriu}(\mathbf{U_D} - \mathbf{B}),$$

or as

$$\mathbf{LD} - \mathbf{B} \approx \mathit{btril}(\mathbf{LD} - \mathbf{B}),$$

where *btriu* and *btril* denote the block upper and block lower triangular parts (including the main diagonal), respectively. Putting the two approximation steps together, we obtain updated preconditioners in the form

$$\mathbf{M}^+ = \mathbf{L}\big(\mathbf{U_D} - \mathit{btriu}(\mathbf{B})\big) \tag{6}$$

and

$$\mathbf{M}^+ = \big(\mathbf{LD} - \mathit{btril}(\mathbf{B})\big)\mathbf{U}. \tag{7}$$

They can be obtained very cheaply. They ask only for subtracting block triangular parts of $\mathbf{A}$ and $\mathbf{A}^+$ (and for saving the corresponding block triangular part of $\mathbf{A}$). In addition, as the sparsity patterns of the factors from the BILU(0) factorization and from the block triangular parts of $\mathbf{A}$ (and $\mathbf{A}^+$) are identical, both backward and forward substitution with the updated preconditioners are as cheap as with the frozen preconditioner $\mathbf{LU_D} = \mathbf{LDU}$.

It is clear from the two approximations we make, that the distance of the proposed updated preconditioners (6) and (7) to the ideal preconditioner is mainly influenced by the following two properties. The first is closeness of $\mathbf{L}$ or $\mathbf{U}$ to the identity. If matrices have a strong diagonal, the diagonal dominance is in general inherited by the factors $\mathbf{L}$ and $\mathbf{U}$ [4,2], yielding reasonable approximations of the identity. The second property that helps in approximating the ideal preconditioner is a block triangular part containing significantly more relevant information than the other part. In one of our model problems we emphasize one triangular part by using a numbering of grid cells corresponding to the direction of the flow characteristics. Summarizing, one may expect updates of the form (6) or (7) to be accurate whenever *btril*($\mathbf{B}$) or *btriu*($\mathbf{B}$) is a useful approximation of $\mathbf{B}$ and when the factor $\mathbf{L}$ or $\mathbf{U}$ is close to the identity matrix. The following lemma suggests that under the mentioned circumstances, the updates have the potential to be more accurate than the frozen or any other (possibly recomputed) preconditioner for $\mathbf{A}^+$.

Lemma 1. *Let* $\|\mathbf{A} - \mathbf{LDU}\| = \varepsilon\|\mathbf{A}\| < \|\mathbf{B}\|$ *for some* $\varepsilon > 0$. *Then the preconditioner from* (7) *satisfies*

$$\|\mathbf{A}^+ - \mathbf{M}^+\| \leqslant \frac{\|\mathbf{U}\|\|\mathit{bstriu}(\mathbf{B})\| + \|\mathbf{U} - \mathbf{I}\|\|\mathbf{B}\| + \varepsilon\|\mathbf{A}\|}{\|\mathbf{B}\| - \varepsilon\|\mathbf{A}\|} \cdot \|\mathbf{A}^+ - \mathbf{LDU}\|, \tag{8}$$

where bstriu denotes the block strict upper triangular part.

This result is a straightforward modification of Lemma 2.1 in [9]; a similar statement can be obtained for updates of the form (6). Having a reference preconditioner $\mathbf{LDU}$ which is not too weak we may assume that $\varepsilon\|\mathbf{A}\|$ is small. Then the multiplication factor before $\|\mathbf{A}^+ - \mathbf{LDU}\|$ in (8) is dominated by the expression $\|\mathbf{U}\|\frac{\|\mathit{bstriu}(\mathbf{B})\|}{\|(\mathbf{B})\|} + \|\mathbf{U} - \mathbf{I}\|$, which may become smaller than one when *btril*($\mathbf{B}$) contains most of $\mathbf{B}$ and when $\mathbf{U}$ is close to the identity matrix. It is possible to show that also the stability of the updates benefits from situations where *btril*($\mathbf{B}$) contains most of $\mathbf{B}$ and where $\mathbf{U}$ is close to identity. In our context, the stability is measured by the distance of the preconditioned matrix to identity. This conforms to the treatment of the stability in [8]. Note that the problem of stability in ILU-type of preconditioners was introduced in the classical paper [11]. It was shown in [3] how this problem can be alleviated by some matrix reorderings. Theorem 2.2 in [9], which addresses this stability, can easily be adopted for our case with preconditioning from the right instead of from the left and with block-wise factorization.

The next result is more specific to the situation we are interested in here. It presents a simple sufficient condition for superiority of the update in the case where the frozen preconditioner is a BILU(0) factorization. The result exploits the fact that the BILU(0) preconditioner is an exact decomposition with the sparsity pattern of the matrix it preconditions. It is formulated here for the update (6), but has, of course, an analogue for (7). The matrix $\mathbf{E}$ denotes the error $\mathbf{E} \equiv A - \mathbf{LDU}$ of the BILU(0) preconditioner and $\|\cdot\|_F$ stays for the Frobenius norm.

Lemma 2. *If*

$$\sqrt{\|\mathbf{E}\|_F^2 + \big\|\mathit{bstril}(\mathbf{B})\big\|_F^2} < \frac{1 - \|\mathbf{I} - \mathbf{L}\|_F^2}{2\|\mathbf{I} - \mathbf{L}\|_F}\big\|\mathit{btriu}(\mathbf{B})\big\|_F, \tag{9}$$

where bstril denotes the block strict lower triangular part of a matrix, then the accuracy of the updated preconditioner $\|\mathbf{A}^+ - \mathbf{L}(\mathbf{DU} - \mathit{btriu}(\mathbf{B}))\|_F$ *is higher than the accuracy* $\|\mathbf{A}^+ - \mathbf{LDU}\|_F$ *of the frozen preconditioner.*

Proof. We have

$$\begin{aligned}\left\|\mathbf{A}^+ - \mathbf{L}\big(\mathbf{DU} - btriu(\mathbf{B})\big)\right\|_F^2 &= \left\|\mathbf{A} - \mathbf{LDU} - \mathbf{B} + \mathbf{L}\cdot btriu(\mathbf{B})\right\|_F^2 \\ &= \left\|\mathbf{E} - bstril(\mathbf{B}) - (\mathbf{I}-\mathbf{L})btriu(\mathbf{B})\right\|_F^2 \leqslant \left(\left\|\mathbf{E} - bstril(\mathbf{B})\right\|_F + \left\|(\mathbf{I}-\mathbf{L})btriu(\mathbf{B})\right\|_F\right)^2 \\ &= \left\|\mathbf{E} - bstril(\mathbf{B})\right\|_F^2 + 2\left\|\mathbf{E} - bstril(\mathbf{B})\right\|_F\left\|(\mathbf{I}-\mathbf{L})btriu(\mathbf{B})\right\|_F + \left\|(\mathbf{I}-\mathbf{L})btriu(\mathbf{B})\right\|_F^2.\end{aligned}$$

Note that the sparsity patterns of $\mathbf{A}$ and $\mathbf{E}$ are disjoint. Hence, with the assumption (9),

$$\begin{aligned}&\left\|\mathbf{E} - bstril(\mathbf{B})\right\|_F^2 + 2\left\|\mathbf{E} - bstril(\mathbf{B})\right\|_F\left\|(\mathbf{I}-\mathbf{L})btriu(\mathbf{B})\right\|_F + \left\|(\mathbf{I}-\mathbf{L})btriu(\mathbf{B})\right\|_F^2 \\ &\leqslant \left\|\mathbf{E} - bstril(\mathbf{B})\right\|_F^2 + 2\left\|\mathbf{E} - bstril(\mathbf{B})\right\|_F\left\|(\mathbf{I}-\mathbf{L})\right\|_F\|btriu(\mathbf{B})\|_F + \left\|(\mathbf{I}-\mathbf{L})\right\|_F^2\left\|btriu(\mathbf{B})\right\|_F^2 \\ &< \left\|\mathbf{E} - bstril(\mathbf{B})\right\|_F^2 + \left(\mathbf{1} - \|\mathbf{I}-\mathbf{L}\|_F^2\right)\left\|btriu(\mathbf{B})\right\|_F^2 + \left\|(\mathbf{I}-\mathbf{L})\right\|_F^2\left\|btriu(\mathbf{B})\right\|_F^2 \\ &< \left\|\mathbf{E} - bstril(\mathbf{B})\right\|_F^2 + \left\|btriu(\mathbf{B})\right\|_F^2 \\ &= \|\mathbf{A}^+ - \mathbf{LDU}\| - \left\|btriu(\mathbf{B})\right\|_F^2 + \left\|btriu(\mathbf{B})\right\|_F^2 = \|\mathbf{A}^+ - \mathbf{LDU}\|. \qquad \square\end{aligned}$$

Lemmas 1 and 2 may be used in practice to predict what type of update, (6) or (7), will perform better. For example, one may compare the multiplication factor before $\|\mathbf{A}^+ - \mathbf{LDU}\|$ in (8) when using (6) or (7) or compare the differences between the left and right hand side in (9) for the choice (6) and the choice (7). However, inequality (9) cannot be satisfied when the numerator is negative, which is very probable in large dimensions. Also, our experience is that the factor before $\|\mathbf{A}^+ - \mathbf{LDU}\|$ in (8) is larger than one in many cases.

Because of this we present a result which is based on the same idea as (9) but it is stronger. The price for getting a significantly tighter bound is using a less transparent assumption. The result also reveals that the quality of the updates is influenced by further, and more subtle properties than only by closeness of triangular factors to the identity matrix and by the dominance of one triangular part of $\mathbf{B}$.

Lemma 3. *Let*

$$\rho = \frac{\|btril(\mathbf{B})(\mathbf{I}-\mathbf{U})\|_F\left(2\cdot\|\mathbf{E} - bstriu(\mathbf{B})\|_F + \|btril(\mathbf{B})(\mathbf{I}-\mathbf{U})\|_F\right)}{\|btril(\mathbf{B})\|_F^2} < 1.$$

Then the accuracy $\|\mathbf{A}^+ - (\mathbf{LD} - btril(\mathbf{B}))\mathbf{U}\|_F$ *of the updated preconditioner* (7) *is higher than the accuracy of the frozen preconditioner* $\|\mathbf{A}^+ - \mathbf{LDU}\|_F^2$ *with*

$$\left\|\mathbf{A}^+ - \big(\mathbf{LD} - btril(\mathbf{B})\big)\mathbf{U}\right\|_F \leqslant \sqrt{\|\mathbf{A}^+ - \mathbf{LDU}\|_F^2 - (1-\rho)\left\|btril(\mathbf{B})\right\|_F^2}. \tag{10}$$

Proof. We have, by assumption,

$$\begin{aligned}\left\|\mathbf{A}^+ - \big(\mathbf{LD} - btril(\mathbf{B})\big)\mathbf{U}\right\|_F^2 &= \left\|\mathbf{A} - \mathbf{LDU} - \mathbf{B} + btril(\mathbf{B})\mathbf{U}\right\|_F^2 \\ &= \left\|\mathbf{E} - bstriu(\mathbf{B}) + btril(\mathbf{B})(\mathbf{I}-\mathbf{U})\right\|_F^2 \\ &\leqslant \left(\left\|\mathbf{E} - bstriu(\mathbf{B})\right\|_F + \left\|btril(\mathbf{B})(\mathbf{I}-\mathbf{U})\right\|_F\right)^2 \\ &= \left\|\mathbf{E} - bstriu(\mathbf{B})\right\|_F^2 + \rho\left\|btril(\mathbf{B})\right\|_F^2.\end{aligned}$$

Because the sparsity patterns of $\mathbf{A}$ and $\mathbf{E}$ are disjoint,

$$\left\|\mathbf{E} - bstriu(\mathbf{B})\right\|_F^2 + \left\|btril(\mathbf{B})\right\|_F^2 = \|\mathbf{E}\|_F^2 + \|\mathbf{B}\|_F^2 = \|\mathbf{E} - \mathbf{B}\|_F^2 = \|\mathbf{A}^+ - \mathbf{LDU}\|_F^2.$$

Hence

$$\left\|\mathbf{E} - bstriu(\mathbf{B})\right\|_F^2 + \rho\left\|btril(\mathbf{B})\right\|_F^2 = \|\mathbf{A}^+ - \mathbf{LDU}\|_F^2 - (1-\rho)\left\|btril(\mathbf{B})\right\|_F^2. \qquad \square$$

With (10), the value of ρ may be considered a measure for the superiority of the updated preconditioner over the frozen preconditioner. However, interpretation of the value of ρ is not straightforward. We may write ρ as

$$\rho = \left(\frac{\|btril(\mathbf{B})(\mathbf{I}-\mathbf{U})\|_F}{\|btril(\mathbf{B})\|_F} \right)^2 + 2\frac{\|\mathbf{E} - bstriu(\mathbf{B})\|_F}{\|btril(\mathbf{B})\|_F^2}, \tag{11}$$

where the ratio

$$\frac{\|btril(\mathbf{B})(\mathbf{I}-\mathbf{U})\|_F}{\|btril(\mathbf{B})\|_F} \tag{12}$$

shows an interesting dependence of ρ on the extent to which $btril(\mathbf{B})$ is reduced after its postmultiplication by $(\mathbf{I}-\mathbf{U})$. This is something slightly different from the dependence of the quality of the update on the closeness of $\mathbf{U}$ to identity. In general, also the second term in (11) should be taken into account; only when the lower triangular part clearly dominates and when $\mathbf{LDU}$ is a powerful factorization, one may concentrate on (12). Computation of ρ is not feasible in practice because of the expensive product in $\|btril(\mathbf{B})(\mathbf{I}-\mathbf{U})\|_F$ but it offers some insight in what really influences the quality of the update. As the proof of the lemma uses only one inequality, one may expect (10) to be a tight bound. We confirm this in the section with numerical experiments.

We will now describe how we exploit updated preconditioners of the form (6) and (7) in the solution process of the problems introduced in the previous section. A first issue is the choice between (6) and (7). We can use some of the previous lemmas to make this choice but we prefer simpler strategies. Just as the ideal preconditioner is approximated in two steps, there are basically two types of simple criteria that can be used. The first criterion compares the closeness of the factors to identity, namely the norms $\|\mathbf{L}-\mathbf{I}\|$ and $\|\mathbf{U}-\mathbf{I}\|$. If the former norm is smaller, then we may expect the approximation made in (4) is better than the one in (5) and we prefer to update the upper triangular part of the decomposition as given in (6); if, on the contrary, $\mathbf{U}$ is closer to identity in some norm, we update the lower triangular part according to (7). Note that a factor close to identity also leads to stable back or forward substitution with the factor. Therefore, an important consequence of choosing the factor which is closest to identity is that we keep, in the update, the more stable part of the initial decomposition. Due to the lack of diagonal dominance in our applications, stability of the factors is a relevant issue. We call this criterion the *stable update criterion*. On the other hand, it is clear that the quality of the approximation $\mathbf{U_D} - btriu(\mathbf{B})$ of $\mathbf{U_D} - \mathbf{B}$ (or $\mathbf{LD} - btril(\mathbf{B})$ of $\mathbf{LD} - \mathbf{B}$) may have a decisive influence on the power of the preconditioner. The second criterion consists of comparing of $\|btril(B)\|$ and $\|btriu(\mathbf{B})\|$. We assume the most important information is contained in the dominating block triangular part and therefore we update with (6) if $btriu(\mathbf{B})$ dominates $btriu(\mathbf{B})$ in an appropriate norm. Otherwise, (7) is used. This rule is denoted by *information flow criterion*. Note that in our implementation we always used the Frobenius norm to evaluate the criteria.

Our model problems lead to systems with a block structure and for efficiency reasons, this block structure should be exploited whenever possible. In order to solve linear systems blockwise and, in particular, work with BILU(0) decompositions, we have adapted the original updating technique to updates of the form (6) and (7). Blockwise decompositions, however, make the switch between (6) and (7) a slightly more complicated than in the case of classical pointwise decompositions. Using the update (6) is straightforward but note that in order to obtain $\mathbf{U}$ and to apply (7) we need to scale $\mathbf{U_D}$ by $\mathbf{D}^{-1}$, as we explained in Section 3.1. Scaling with inverse block diagonal matrices does have, in contrast with inverse diagonal matrices, some influence on overall performance and should be avoided as much as possible. Note that our stable update criterion compares $\|\mathbf{L}-\mathbf{I}\|$ with $\|\mathbf{U}-\mathbf{I}\|$ where both factors $\mathbf{L}$ and $\mathbf{U}$ have a block diagonal consisting of identity blocks. This means that in order to use the criterion we need to scale $\mathbf{U_D}$, even if the criterion decides for (6) and scaling would not have been necessary. We may circumvent this possible inefficiency by considering $\mathbf{U_D}$ and $\mathbf{LD}$ instead of $\mathbf{U}$ and $\mathbf{L}$. More precisely, we would compare $\|\mathbf{D}-\mathbf{U_D}\|$ with $\|\mathbf{LD}-\mathbf{D}\|$. We call this third criterion the *unscaled stable update criterion*.

A related issue is the frequency of deciding about the update type based on the chosen criterion. On one hand, there may be important differences in the performance of (6) and (7); on the other hand, switching between the two types implies some additional costs like, for instance, storage of both triangular parts of $\mathbf{B}$. Consequently, we believe that the criterion query should not be repeated too often. We adopted the following strategy. After every recomputation of the BILU(0) decomposition, which takes place periodically, we perform one query and then use the chosen type of update throughout the whole period. With the information flow criterion we compare $\|btril(\mathbf{B})\|$ with $\|btriu(\mathbf{B})\|$ for the first difference matrix $\mathbf{B}$ generated after recomputation, i.e. just before solving the system following the system for which we used a new BILU(0) decomposition. For the two stable update criteria we may decide immediately which

update type should be used for the next couple of iterations as soon as the new BILU(0) decomposition was computed. Note that as soon as the update type is chosen, we need to store only one triangular part of the old reference matrix $\mathbf{A}$ (and two triangular factors of the reference decomposition).

Another property of the applications we are interested in here, is that the solution process typically contains heavily instationary phases followed by long nearly stationary phases. This is reflected by parts of the sequence of linear systems with large entries in the difference matrices and other parts where system matrices are very close. Obviously, in the latter parts we may expect a frozen preconditioner to be powerful for many subsequent systems. Our experiments confirm this: In stationary phases we typically observe a deterioration of only 2 to 5 iterations with respect to the iterations needed to solve the system for which the frozen preconditioner was used. Updating the frozen preconditioner in these cases would be counterproductive; it would add some overhead which cannot be compensated by the few savings of iterations. In fact, in these cases there is even a risk that updates produce more iterations, especially when the frozen preconditioner is particularly stable. We therefore apply a very simple technique to avoid unnecessary updating. We start every period by freezing the preconditioner. Denote the number of iterations of the linear solver needed to solve the first system of the period by $iter_0$. If for the $(j+1)$st system the corresponding number of iterations $iter_j$ satisfies

$$iter_j > iter_0 + k, \tag{13}$$

with some threshold $k \in \mathbb{N}$, then we use updates for all remaining systems of the period. In accordance with our observations, we used $k = 3$. To get a clearer impression of the code decisions to be made we have added a flow diagram. Here, p denotes the recomputation period and $m = 0, 1, 2, \ldots$.

Flow diagram—preconditioner update decisions after recomputation.

- Compute a block ILU decomposition $\mathbf{LU_D}$ of $\mathbf{A^{(mp)}}$, set $type = frozen$
- If a stable update criterion is used
  - Perform criterion query, scale $\mathbf{U_D}$ if updates of the form (7) are to be used
- Solve $\mathbf{A^{(mp)}x^{(mp)}} = \mathbf{b^{(mp)}}$ with $type$ preconditioner in $iter_0$ iterations
- If the information flow criterion is used
  - Perform criterion query, scale $\mathbf{U_D}$ if updates of the form (7) are to be used
- For $i = 1, 2, \ldots, p-1$
  - Solve $\mathbf{A^{(mp+i)}x^{(mp+i)}} = \mathbf{b^{(mp+i)}}$ with $type$ preconditioner in $iter_i$ iterarions
  - If $type = frozen$
    - If $iter_i > iter_0 + k$
      - Set $type = updated$

4. Numerical experiments

In this section we demonstrate the behavior of the update technique on some well known steady state test cases. The corresponding linear equation systems are solved until the initial residual has dropped by a factor of 10^7. We always compare periodic refactorization without updating to periodic refactorization with updating, where also the three criteria for deciding whether to use upper or lower updating are compared. The total number of BiCGSTAB iterations as well as the total CPU time for the whole run are recorded. Our primary indicator to evaluate performance

is the CPU time, as a small number of BiCGSTAB iterations may be due to the block preconditioner that takes tremendous amount of computational time. All computations were performed on a Pentium IV with 2.4 GHz.

4.1. Supersonic flow past a cylinder

The first model problem is frontal flow at Mach 10 around a cylinder, which leads to a steady state. 3000 steps of the implicit Euler method are performed. The grid consists of 20 994 points, whereby only a quarter of the domain is discretized, and system matrices are of dimension 83 976. The number of nonzeroes is about 1.33×10^6 for all matrices of the sequence. For the initial data, freestream conditions are used. Thus, in the beginning, a strong shock detaches from the cylinder, which then slowly moves backward through the domain until reaching the steady state position. Therefore, the linear systems are changing only very slowly during the last 2500 time steps and all important changes take place in the initial phase of 500 time steps. The initial CFL number is 5, which is increased up to 7 during the iteration. The solution is shown in Fig. 2.

As the flow is supersonic, the characteristics point mostly in one direction. The performance of the linear equation solver can be improved by choosing a numbering of the grid cells that respects the direction of the flow, thereby making the matrix more triangular in nature. This is achieved by numbering first the cells from the inflow boundary, then the cells in direction of the characteristics and by continuing in this manner repeatedly, see [16]. Renumbering reduces the total number of BiCGSTAB iterations by about thirty percent. Furthermore, dominance of one of the two triangular parts is exactly the situation in which we expect the update technique to work well. Recall that Lemmas 1, 2 and 3 all suggest that the updated preconditioner is favorably influenced by matrices with a dominating triangular part. In Fig. 2 excellent performance of the updates is shown for the initial unsteady phase of the first 500 time steps. As subsequent linear systems change heavily, frozen preconditioners produce rapidly deteriorating numbers of BiCGSTAB iterations (with decreasing peaks demonstrating the convergence to steady state). Updating, on the other hand, yields a nearly constant number of iterations per time step. The recomputing period here is thirty and the criterion used is the stable update criterion but other periods and criteria give a similar result. With freezing, 5380 BiCGSTAB iterations are performed in this part of the solution process, while the same computation with updating needs only 2611 iterations.

In Table 1 we explain the superior performance of the updates with the quantities from Lemma 3 for the very first time steps; they demonstrate the general trend for the whole instationary phase. Here, $M^{(i)}$ denotes the update (7) for the ith linear system. As the upper bound (10) on the accuracy of the updates is very tight, we conclude that in this problem the power of the updates is essentially due to the small values of ρ.

In Table 2 we display the performance of the updates for the whole sequence. To evaluate the results, first note that the reduction of the BiCGSTAB iterations happens primarily in the first 500 time steps. After 500 time steps,

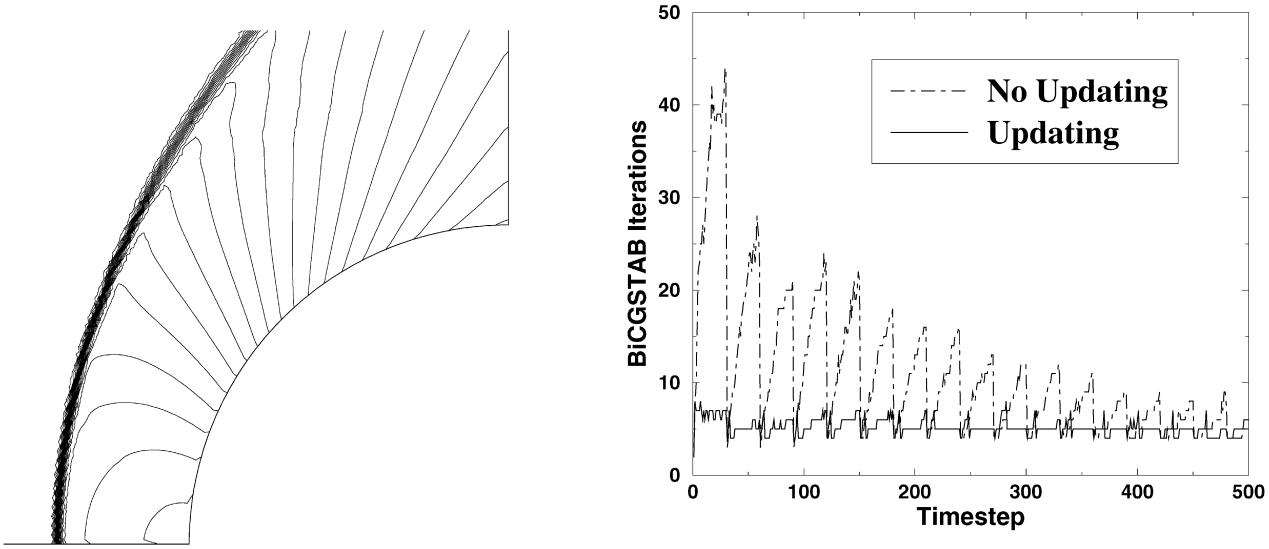


Fig. 2. Pressure isolines (left) and BiCGSTAB iterations per time step (right) for the cylinder problem.

Table 1
Accuracy of the preconditioners and theoretical bounds

i	$\|A^{(i)} - LDU\|_F$	$\|A^{(i)} - M^{(i)}\|_F$	Bound from (10)	ρ from (10)
2	37.454	34.277	36.172	0.571
3	37.815	34.475	36.411	0.551
4	42.096	34.959	36.938	0.245
5	50.965	35.517	37.557	0.104
6	55.902	36.118	38.308	0.083

Table 2
Total iterations and CPU times for supersonic flow example

Period	No updating		Stable update		Unscaled stable update		Information flow	
	Iter.	CPU in s	Iter.	CPU in s	Iter.	CPU in s	Iter.	CPU in s
10	10683	7020	11782	7284	11782	7443	11782	7309
20	12294	6340	12147	6163	12147	6300	12147	6276
30	13787	7119	12503	5886	12503	5894	12503	5991
40	15165	6356	12916	5866	12916	5835	12916	5856
50	16569	6709	13139	5786	13139	5715	13139	5740

freezing is a very efficient strategy and actually gains again on updating. Thus the visual success of updating is somewhat damped by the long stationary tail of this model problem. The different updating strategies lead to nearly identical results, whereby the stable update criterion is the best, except for the last two periods. As expected, the update criterions all choose to update the lower triangular part according to (7), as the upper triangular part is close to identity due to the numbering of the unknowns and the high Mach number. Therefore, they all obtain the same iteration numbers. Updating is clearly better than freezing if the recomputing period is at least 20. For recomputing periods of 30 or greater, the performance of the updating strategy does not much depend on the period. The CPU time is decreased by about 10% in general; with the recomputing period 50 it reaches up to 20%. For longer recomputing periods, the number of iterations is reduced by even more than 20%. For the period 10 the frozen preconditioner does not deteriorate very much during the periods and achieves lower overall numbers of iterations (and timings) than any updates. This must be caused by the fact that the frozen preconditioner is more stable than the updates. However, the recomputing period 10 is easily beaten by longer periods. If the BILU(0) decomposition would have been recomputed in *every* step, only 11 099 BiCGSTAB iterations would be needed, but 28 583 seconds of CPU time.

4.2. Flow past a NACA0012 airfoil

The second model problem corresponds to the NACA0012 profile at an angle of attack of two degrees on a grid with 4605 cells at different Mach numbers. System matrices are of dimension 18 420 and the number of nonzeroes is about 5×10^5 for all matrices of the sequence. For the initial data, freestream conditions are used.

At first we consider a reference Mach number of $M = 0.8$. 1000 steps of the implicit Euler method are performed. The initial CFL number is 5, which is increased up to 30 during the process. For the solution, see Fig. 3 (left). Transition to steady state is such that after the shock on the airfoil has formed, the rate of convergence slows down, even though the CFL number is increased. Similarly as in the supersonic model, the equation systems differ much from step to step at first, but are very close towards the end. In fact, this behavior is here even more extreme: With decisions based on (13), updating is applied during the very first period only. To illustrate this, Fig. 3 (right) compares, for recomputation with a period of 30 time steps, classical freezing with our strategy. Clearly, increasing BiCGSTAB iteration numbers of the frozen preconditioner can be corrected with the updates. But after the first period, there is no need to correct anymore.

The entire process is shown in Table 3. As we can see, the number of iterations decreases if the recomputation period is shortened. This is not true for the CPU time, as recomputations are costly. For the strategy without updating, the CPU time decreases at first, but increases again, as the benefit of fewer recomputations is balanced by the increase in BiCGSTAB iterations. As for the different updating strategies, all lead to both fewer iterations and shorter computing times. As we explained before, the reduction of iterations must be solely due to the very first time steps

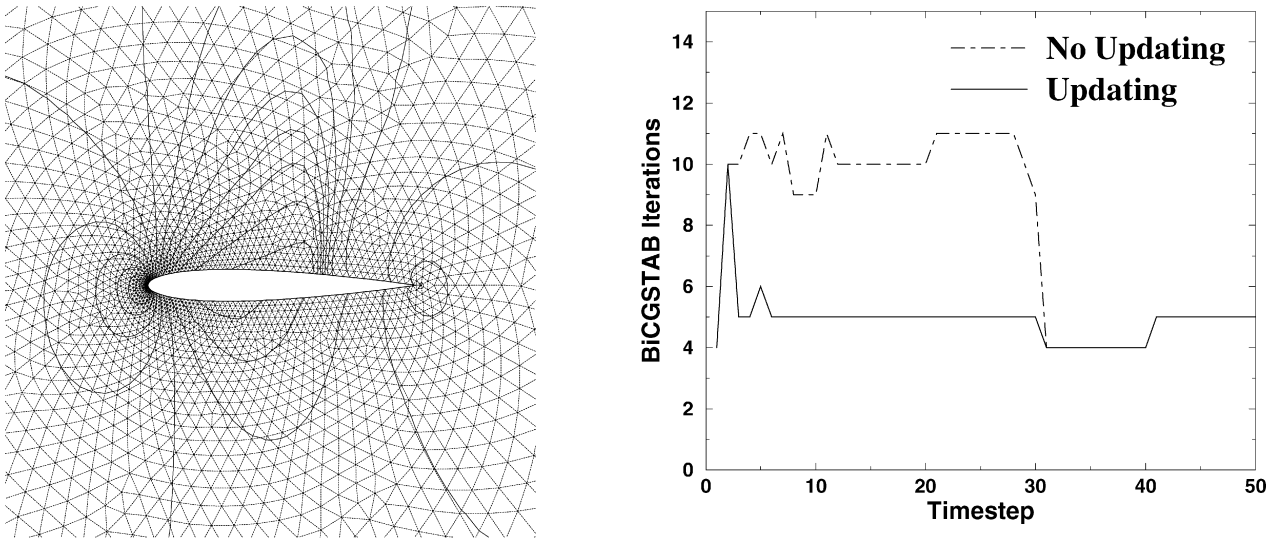


Fig. 3. Pressure isolines and grid (left) and BiCGSTAB iterations per time step (right) for NACA profile with Mach 0.8.

Table 3
Total iterations and CPU times for transonic flow example

Period	No updating		Stable update		Unscaled stable update		Information flow	
	Iter.	CPU in s	Iter.	CPU in s	Iter.	CPU in s	Iter.	CPU in s
10	5375	543	5336	498	5336	494	5336	483
20	5454	497	5364	469	5364	468	5364	459
30	5526	491	5379	464	5379	467	5379	453
40	5558	491	5411	456	5411	462	5411	452
50	5643	525	5413	466	5413	470	5413	448

where updates are applied. The information flow criterion provides the fastest results, whereas the stable update criterion and the unscaled stable update criterion lead to somewhat higher total timings, but still faster than without any updates. All three criterions lead to an identical number of BiCGSTAB iterations, because they always choose the same triangular part to update. If the BILU(0) decomposition would have been recomputed in every step, only 5333 BiCGSTAB-iterations would be needed, but 964 seconds of CPU time. Thus the number of iterations with updating often comes close to the number with refactorization in every single step. The differences in CPU time come from the cost of selecting the appropriate triangular part and all in all, the computation of the steady state is improved up by about 7 to 15%. Note that again, the CPU time depends less on the choice of the recomputation period with updates than is the case without updating.

In the last test case we use a Mach number of $M = 0.001$. This problem is much more stiff than the transonic problem. Consequently, the linear systems are harder to solve. Furthermore, for the same CFL number, the time steps should be much smaller due to the larger maximum eigenvalues of the involved matrices. We computed 750 time steps, starting with the CFL number of 0.5, which was increased to its final value equal to two. For the solution, see Fig. 4, for the comparison of updating techniques see Table 4. In this case, the linear systems do not differ very much among the time steps, not even in the beginning. Thus, the freezing strategy works well and the number of iterations needed increases very slowly in one recomputation cycle. Therefore, even if updating is used, the criterion (13) is seldom fulfilled and the updating strategy has only a small effect in decreasing the iteration numbers, but essentially none on the CPU time. Nevertheless, it is not worse than the classic strategy, which is mainly due to the inclusion of criterion (13): Otherwise, the method would compute an update in every step to no effect. Note that if the BILU(0) decomposition would have been recomputed in every step, 19 609 BiCGSTAB-iterations would be needed, but 1437 seconds of CPU time.

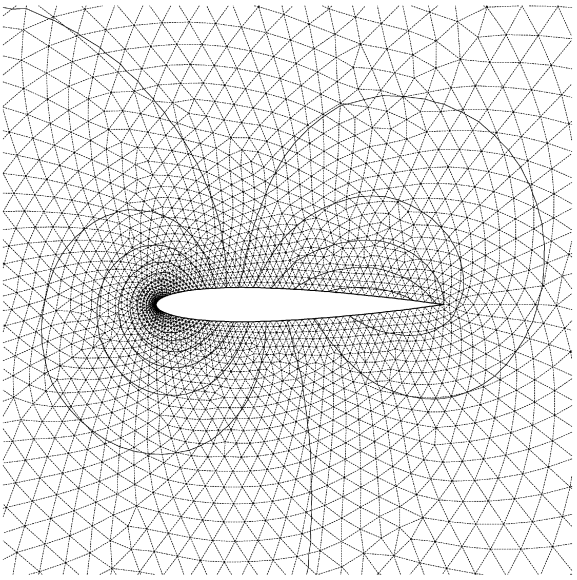


Fig. 4. Pressure isolines and Grid for NACA profile at Mach 0.001.

Table 4
Total iterations and CPU times for low Mach flow example

Period	No updating		Stable update		Unscaled stable update		Information flow	
	Iter.	CPU in s	Iter.	CPU in s	Iter.	CPU in s	Iter.	CPU in s
10	19444	1189	19288	1158	19398	1121	19289	1129
20	19584	1105	19492	1135	19451	1117	19375	1094
30	19641	1144	19412	1122	19531	1158	19544	1112
40	19622	1104	19521	1112	19594	1114	19523	1107
50	19622	1127	19265	1129	19339	1086	19396	1139

5. Conclusions

We employed an updating method for block ILU preconditioners for sequences of nonsymmetric linear systems in the context of compressible flow. The updating method was motivated by the need to improve frozen preconditioners in order to obtain preconditioners similarly powerful as if they would have been recomputed. For the model problems considered here we showed that as soon as the frozen preconditioners yield high numbers of iterations of the linear solver, the updates indeed succeed in reducing the number to the normal level. Whereas the derivation of the updates assumes diagonal dominance of system matrices, the present experiments imply the technique is efficient with rather poor diagonal dominance as well. Note that the success of the new strategy may be significantly enhanced if the time for recomputations becomes prohibitive, which was not our case.

Based on the number of Krylov subspace method iterations, our implementation decides whether updating is necessary. In this way we obtained a preconditioning strategy that is faster than the standard strategy of periodic recomputing for well-known test cases and it is even close to recomputing in every step with respect to iteration numbers. In contrast to periodic recomputations without updates, our method is rather insensitive to the chosen recomputation period.

The method is particularly successful in the phases where the solution process exhibits some kind of instationary behavior and thus it is promising for the computation of instationary flows. In our tables we willingly chose to display results for the whole solution process including long stationary phases of the problems. If we would restrict ourselves to the phases where the updates were actually applied, the results would be even more convincing.

References

[1] J. Baglama, D. Calvetti, G.H. Golub, L. Reichel, Adaptively preconditioned GMRES algorithms, SIAM J. Sci. Comput. 20 (1998) 243–269.
[2] M. Benzi, D. Bertaccini, Approximate inverse preconditioning for shifted linear systems, BIT 43 (2003) 231–244.

[3] M. Benzi, D.B. Szyld, A. van Duin, Orderings for incomplete factorization preconditioners of nonsymmetric problems, SIAM J. Sci. Comput. 20 (1999) 1652–1670.
[4] M. Benzi, M. Tůma, Orderings for factorized sparse approximate inverse preconditioners, SIAM J. Sci. Comput. 21 (2000) 1851–1868.
[5] L. Bergamaschi, R. Bru, A. Martínez, M. Putti, Quasi-Newton preconditioners for the inexact Newton method, ETNA 23 (2006) 76–87.
[6] D. Bertaccini, Efficient preconditioning for sequences of parametric complex symmetric linear systems, Electron. Trans. Numer. Math. 18 (2004) 49–64.
[7] A. Chapman, Y. Saad, L. Wigton, High-order ILU preconditioners for CFD problems, Int. J. Numer. Methods Fluids 33 (2000) 767–788.
[8] E. Chow, Y. Saad, Experimental study of ILU preconditioners for indefinite matrices, J. Comp. Appl. Math. 86 (1997) 387–414.
[9] J. Duintjer Tebbens, M. Tůma, Efficient preconditioning of sequences of nonsymmetric linear systems, SIAM J. Sci. Comput. 19 (2007) 1918–1941.
[10] H.C. Elman, A. Ramage, Fourier analysis of multigrid for the two-dimensional convection–diffusion equation, BIT Numer. Math., online version (May, 2006).
[11] H.C. Elman, A stability analysis of incomplete LU factorization, Math. Comp. 47 (1986) 191–218.
[12] R.J. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge Texts in Applied Mathematics, Cambridge University Press, Cambridge, 2002.
[13] D. Loghin, D. Ruiz, A. Touhami, Adaptive preconditioners for nonlinear systems of equations, J. Comput. Appl. Math. 189 (2006) 326–374.
[14] D. Loghin, D. Ruiz, A. Touhami, Asymptotic based preconditioning technique for low Mach number flows, Z. Angew. Math. Mech. 83 (2003) 3–25.
[15] A. Meister, T. Sonar, Finite-volume schemes for compressible fluid flow, Surveys Math. Indust. 8 (1998) 1–36.
[16] A. Meister, C. Vömel, Efficient preconditioning of linear systems arising from the discretization of hyperbolic conservation laws, Adv. Comput. Math. 14 (2001) 49–73.
[17] G. Meurant, On the incomplete Cholesky decomposition of a class of perturbed matrices, SIAM J. Sci. Comput. 23 (2001) 419–429.
[18] J. Morales, J. Nocedal, Automatic preconditioning by limited-memory quasi-Newton updates, SIAM J. Opt. 10 (2000) 1079–1096.
[19] M.L. Parks, E. de Sturler, G. Mackey, D.D. Johnson, S. Maiti, Recycling Krylov subspaces for sequences of linear systems, Technical Report UIUCDCS-R-2004-2421, University of Illinois, 2004.
[20] M.L. Parks, E. de Sturler, G. Mackey, D.D. Johnson, S. Maiti, Recycling Krylov subspaces for sequences of linear systems, SIAM J. Sci. Comput. 28 (2006) 1651–1674.
[21] Y. Saad, Iterative Methods for Sparse Linear Systems, second ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.
[22] Y. Saad, M.H. Schulz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.
[23] V.V. Shaĭdurov, Multigrid Methods for Finite Elements, Mathematics and its Applications, vol. 318, Kluwer Academic Publishers Group, Dordrecht, 1995. Translated from the 1989 Russian original by N.B. Urusova and revised by the author.
[24] H.A. van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems, SIAM J. Sci. Stat. Comput. 12 (1992) 631–644.
[25] Y. Wada, M.-S. Liou, A flux splitting scheme with high-resolution and robustness for discontinuities, AIAA Paper, 94-0083, 1994.
[26] C.-T. Wu, H.C. Elman, Analysis and comparison of geometric and algebraic multigrid for convection–diffusion equations, SIAM J. Sci. Comput. 28 (2006) 2208–2228.

NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2010; **17**:997–1019
Published online 9 February 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nla.695

Preconditioner updates for solving sequences of linear systems in matrix-free environment

Jurjen Duintjer Tebbens and Miroslav Tůma*, †

Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 2, 18207 Praha 8, Czech Republic

SUMMARY

We present two new ways of preconditioning sequences of nonsymmetric linear systems in the special case where the implementation is matrix free. Both approaches are fully algebraic, they are based on the general updates of incomplete LU decompositions recently introduced in (*SIAM J. Sci. Comput.* 2007; **29**(5):1918–1941), and they may be directly embedded into nonlinear algebraic solvers. The first of the approaches uses a new model of partial matrix estimation to compute the updates. The second approach exploits separability of function components to apply the updated factorized preconditioner via function evaluations with the discretized operator. Experiments with matrix-free implementations of test problems show that both new techniques offer useful, robust and black-box solution strategies. In addition, they show that the new techniques are often more efficient in matrix-free environment than either recomputing the preconditioner from scratch for every linear system of the sequence or than freezing the preconditioner throughout the whole sequence. Copyright © 2010 John Wiley & Sons, Ltd.

Received 11 December 2008; Revised 16 November 2009; Accepted 4 December 2009

KEY WORDS: preconditioned iterative methods; matrix-free environment; factorization updates; inexact Newton–Krylov methods; incomplete factorizations

1. INTRODUCTION

We consider the solution of sequences of linear systems

$$A^{(i)}x = b^{(i)}, \quad i = 1, \ldots, \tag{1}$$

*Correspondence to: Miroslav Tůma, Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 2, 18207 Praha 8, Czech Republic.
†E-mail: tuma@cs.cas.cz

Contract/grant sponsor: Agency of the Academy of Sciences of the Czech Republic; contract/grant number: IAA100300802
Contract/grant sponsor: Agency of the Academy of Sciences of the Czech Republic; contract/grant number: KJB100300703
Contract/grant sponsor: International Collaboration; contract/grant number: M100300902

where $A^{(i)} \in \mathbb{R}^{n\times n}$ are general nonsingular sparse matrices and $b^{(i)} \in \mathbb{R}^n$ are corresponding right-hand sides. Such sequences arise in numerous industrial and scientific computations, for example, when a system of nonlinear equations is solved by a Newton or Broyden-type method [1, 2]. Krylov subspace methods are among the most successful approaches for solving the linear systems. These methods have the property that the system matrix is needed only in the form of matrix-vector products, and the explicit representation of the matrix is not necessary. If the system matrix is not represented explicitly, we often say that the method is *matrix free*.

It is widely recognized that in most cases of practical interest, Krylov subspace methods must be preconditioned in order to be efficient and robust. However, most of the strong preconditioners either require the system matrix explicitly, or their computation may be rather expensive. In order to reduce the costs of the computation of preconditioners, we may reuse a preconditioner over more systems of the given sequence of systems of linear equations. The quality of the reused preconditioner may be enhanced through updates containing information extracted from the sequence of matrices, or from the previous application of the Krylov subspace method. Owing to the costs that are related to the fact that the system matrix is not given explicitly, and which may be magnified in a parallel computing environment, avoiding frequent recomputations of the preconditioner from scratch seems to be very important in matrix-free environment.

In this paper we address the problem of solving a sequence of general nonsymmetric systems in matrix-free environment by Krylov subspace methods with preconditioners that are based on incomplete LU decompositions and that are updated with general rank-n approximate modifications. In the next paragraph, we briefly summarize the basic lines of previous research on matrix-free preconditioning and on solving sequences of systems of linear equations with preconditioner updates, that is, on the two subproblems which we face. As far as we know, the combination of these subproblems has never been solved in the past. We propose two ways to do this by overcoming both theoretical and practical obstacles. These ways differ by the necessary sizes of intermediate memory and they can be useful in different applications.

Let us first mention several *preconditioning strategies designed for or related to matrix-free environment*. First, the preconditioners can correspond to a discretized operator, which is simpler than the discretized operator for evaluating the sparse system matrix, see, e.g. [3–8]. Successful preconditioners derived directly from the advection–diffusion operators were also proposed for solving problems in applications that provide rather dense Jacobian matrices [9, 10]. All these physics-based preconditioners are often used in matrix-free environment. Second, a preconditioner can be algebraic. The lack of explicit availability of the system matrix then often implies that the preconditioner is rather simple and/or sparse. In some of such situations, the preconditioner is the matrix diagonal or its approximation. In other situations, preconditioning employs more complex stationary iterative methods, fast FFT-based solvers, ADI methods, inner–outer schemes, etc., in order to be easily applied in matrix-free environment. An early important paper which explicitly targets preconditioning in matrix-free environment is [11] with results for a model nonlinear boundary value problem, see also the applications in CFD [12]. For details on the important class of Jacobian-free Newton–Krylov methods (JFNK) that combine a matrix-free approach with nonsymmetric Krylov subspace methods and for modifications of this class, see the overview in [7], but also in [3].

Preconditioner updates used for *solving system sequences* are traditionally based on modifications by matrices of small rank. Early work that uses the Broyden formula to update the preconditioner was introduced in [13]. Other updates based on matrices of small rank were considered in [14–16] and in limited-memory variable metric methods [17] for smooth optimization, to name just of few.

DOI: 10.1002/nla

An important class of algebraically motivated strategies to accelerate the convergence of preconditioned iterative methods is based on constructing or improving the preconditioner by adaptive spectral information obtained directly from the Krylov subspace methods, see, e.g. [18–25]. All of these techniques have a significant potential to be applied in the form of preconditioner updates for solving sequences of systems [26–28]. These strategies are often problem specific, but they are in general compatible with matrix-free implementations.

Although it is possible to analyze the spectral properties of sequences of preconditioned matrices in some important cases, in other situations we know much less. Preconditioner updates of small rank are often restricted to specific classes of problems or nonlinear schemes as well. Therefore, cheap and generally rank-n preconditioner updates are strongly needed. Recently, some new approaches to approximate preconditioner updates were introduced, see, e.g. [29]. The authors in [30] propose approximate diagonal updates to solve parabolic PDEs, see also [31]. Nonsymmetric updates of general incomplete LU decompositions were considered in [32, 33], see also some results in solving real-world problems in [34]. So far, neither of these approaches have addressed the challenges related to preconditioner updates in the matrix-free environment. Some of the mentioned preconditioner updates may not even be compatible with matrix-free environment.

This paper deals with matrix-free algorithms to solve the sequences of linear systems with the general triangular preconditioner updates introduced in [32]. Its adaptation for matrix-free environment is not straightforward and we propose two new approaches to do this. The first of them is based on an efficient matrix estimation technique. More precisely, a new *partial estimation* procedure is proposed. The second matrix-free approach applies the updates via modified forward or backward solves with the preconditioner, inside the iterative method. It is shown that both approaches may be robust in matrix-free environment. The paper is organized as follows. In Section 2 the general algorithmic framework for the updates is briefly summarized and some preliminary terminology for the two basic approaches in matrix-free environment is introduced. The first new approach is presented in detail in Section 3 and the second strategy is described in Section 4. Section 5 discusses numerical experiments for both approaches. The paper is finalized by some concluding remarks.

2. BASIC UPDATE TECHNIQUE AND MATRIX-FREE COMPUTATIONS

The triangular preconditioner updates for nonsymmetric sequences from [32] are defined with the help of the difference between the matrix from the first (*reference*) linear system of a sequence and the *current* system matrix. Let A be the system matrix of the reference system and let A^+ be the current system matrix. If LDU is an incomplete triangular decomposition of A and $B = A - A^+$ is the difference matrix, then the basic triangularly updated preconditioners for the current system are defined as

$$(LD - \mathrm{tril}(B))U \quad \text{or} \quad L(DU - \mathrm{triu}(B)), \tag{2}$$

where tril and triu denote, respectively, lower and upper triangular part of a matrix. We assume that $(LD - \mathrm{tril}(B))$ or $(DU - \mathrm{triu}(B))$, respectively, is nonsingular. Table I compares the costs and memory for one step of a preconditioned iterative method when one recomputes the preconditioner for the linear system of a sequence, denoted as 'Recomp', with the costs and memory when using the first update in (2), denoted as 'Update'. With the recomputation strategy, we denote the approximate LU factors of A^+ by L^+, D^+ and U^+. The table shows that applying the updates

DOI: 10.1002/nla

Table I. Cost comparison between recomputed and updated preconditioning.

Type	Initialization	Solve step	Memory
Recomp	$A^+ \approx L^+D^+U^+$	Solves with L^+D^+, U^+	A^+, L^+D^+, U^+
Update	—	Solves with LD, U, tril(B)	A^+, tril(A), LD, U

is only slightly more expensive and needs a little more memory than recomputing, but of course it saves all factorization costs (initialization). The table provides only a rough comparison; in particular, the amount of overlap between the sparsity patterns of LD and tril(B) may have an important influence on the storage and application costs. When the overlap is important it makes sense to merge LD and tril(B), i.e. compute the difference $LD - \text{tril}(B)$ and perform forward solves with it. Otherwise, separate solves with LD and tril(B) (where only the entries on the main diagonal are merged) are usually more convenient for implementation reasons. In some cases A and A^+ may not have only somewhat different sparsity patterns, but also completely different sizes (i.e. numbers of nonzero entries).

The simple updates (2) can be expected to be efficient when the dominant information in the difference matrix B is contained in one triangular part, like for instance with upwind/downwind finite difference discretizations. However, we showed in [32–34] that the updates, possibly combined with improvements such as specific reorderings, Gauss–Jordan transforms or Gauss–Seidel-type extensions, are beneficial for a much broader spectrum of problems (including some CFD simulations discretized with finite volumes or elements). One remarkable feature is that the updates seem to yield powerful preconditioning not only, as one would expect from the definition (2), when the system matrices are changing slowly. In [34] the updates are most efficient in the transient phase of the simulation where turbulence causes large differences between system matrices. This may be explained by the fact that we take into account part of the large differences through the matrix B in (2). In addition, a part of the structure of A and A^+ is contained in the updates.

The goal of this paper is not so much to show that preconditioner updates of the type (2) can yield a strong acceleration as compared with other preconditioning strategies. For examples where this is the case we refer to [32–34]. The aim of the paper is to introduce techniques that enable efficient usage of these updates in matrix-free environment and that can be applied as a black-box strategy. Because of the latter aim, our experiments often span whole nonlinear processes, even when we know that we could obtain a more profound effect by concentrating on transient phases of the nonlinear solvers. Without loss of generality, we will use the first type of update in (2) in our exposition. In practice, the type of update is chosen dynamically [33, 34].

The updates (2) are based on an incomplete reference factorization LDU. In many applications preconditioners which are simple, as those mentioned in the introduction, and thus naturally matrix free, are not powerful enough due to linear or sub-linear convergence of the corresponding Krylov subspace method. Instead, strong and robust algebraic preconditioners such as some type of incomplete decomposition must be used. They require, however, to be stored explicitly and, in order to be computed, they need the explicit entries of the matrix that they precondition. This means that the system matrix has to be estimated with the help of matrix-vector multiplications (matvecs). Let us shortly describe the standard matrix estimation strategy that is very often used in matrix-free nonlinear solvers or numerical optimization packages.

The matrix estimation problem is the problem to estimate a sparse matrix, given its sparsity structure, by a small number of well-chosen matvecs. Curtis *et al.* [35] were the first to demonstrate

DOI: 10.1002/nla

that all nonzero entries of a sparse matrix can be estimated using a number of matvecs that are often much smaller than the matrix dimension. Estimation of the entries of a generally nonsymmetric matrix B can be formulated as the following problem.

Problem 2.1
Given the sparsity pattern of B find vectors $d_1, \ldots, d_p$ such that for each nonzero entry b_{ij} of B there is a vector d_k, $1 \leqslant k \leqslant p$, satisfying $(Bd_k)_i = b_{ij}(d_k)_j$.

In practice, we need to have p as small as possible so that the number of matvecs needed to obtain all nonzero entries is minimal. Coleman and Moré [36] demonstrated the relation of the matrix estimation Problem 2.1 to the vertex *coloring* of a related graph G by a minimum number of colors. This minimum number is called the chromatic number of G. The so-called direct methods for solving the matrix estimation problem for a matrix B described in Problem 2.1 use as G the intersection graph of B, that is the adjacency graph $G(B^{\mathrm{T}}B)$ of $B^{\mathrm{T}}B$. Note that for an (undirected) adjacency graph $G(C)$ of a square and symmetric matrix C, we define its set of vertices as $V(G(C)) = \{1, \ldots, n\}$ and its set of edges as $E(G(C)) = \{\{i, j\} \mid c_{ij} \text{ is nonzero}\}$. A vertex coloring of the intersection graph labels every vertex with a color in such a way that no two adjacent vertices have the same color. The number of groups of vertices of the related graph with the same color then corresponds to the number of matvecs needed to estimate all entries of the matrix. A recent survey of theoretical results and techniques in this field is Gebremedhin *et al.* [37] where one can find details on many standard matrix estimation strategies.

In matrix-free environment, the factorization *LDU* in (2) has in general been obtained through estimating the reference matrix A, and it is stored explicitly. The update needs in addition *a part* of the difference matrix B, which is *not* given explicitly (only A has been estimated). As the straightforward estimation of the difference matrix by applying Problem 2.1 to A^+ may be expensive, one possible strategy that we propose is based on modified matrix estimation that is reasonably cheap. In this case we use a new enhanced *partial* and *approximate* matrix estimation. This approach is described in Section 3. As in any matrix-free implementation that uses matrix estimation, we will assume that the sparsity pattern of A and A^+ is given. Note that very often the sparsity pattern can be obtained from the subroutine that performs the matvec: The variables involved in the definition of the kth entry of the output vector yield the sparsity pattern of the kth row of the system matrix.

The idea of estimating an implicitly given system matrix, which may or may not be easily available, is frequently used in practice. Straightforward reasons to do so are, for instance, the need to avoid analytical computation of the system matrix, saving the storage costs for the system matrix or simply the fact that the preconditioners which do not need the matrix explicitly, may be weak. However, estimation based on Problem 2.1 needs some intermediate memory, and memory issues are often crucial in matrix-free environment and storage costs related to incomplete factorizations should be kept as low as possible. In Section 4 we describe a strategy to use the triangular updates (2) in matrix-free environment without running any matrix estimation procedure other than for the reference matrix. Then the difference matrix B does not need to be stored. The costs of the technique depend on the degree of *separability* of the function components of the function that performs the matvec. Let us explain in the subsequent paragraph what we mean by separability in our case (cf. the concept of partial separability in optimization, e.g. in [38]).

Consider a Krylov subspace method where the product of the system matrix A with a vector v is replaced by the value of a function $\mathscr{F}$ evaluated at v. We say that $\mathscr{F}$ is separable if the

DOI: 10.1002/nla

evaluation of $\mathscr{F}$ can be easily separated in the evaluation of its function components. That is, the function $\mathscr{F}$ is well separable if the components of the function $\mathscr{F}:\mathbb{R}^n \to \mathbb{R}^n$ can be written as $\mathscr{F}_i:\mathbb{R}^n \to \mathbb{R}$, where $e_i^{\mathrm{T}}\mathscr{F}(v)=\mathscr{F}_i(v)$, and computing $\mathscr{F}_i(v)$ costs about an nth part of the full function evaluation $\mathscr{F}(v)$. Note that in some cases, as they arise in complicated computations based on finite volumes or finite elements, the contributions for each volume or element are computed simultaneously, and in this case, the evaluation of a single function component costs more.

The next section presents our first technique, which is based on solving new matrix estimation problems, to apply the updates in matrix-free environment. Section 4 describes the second technique that fully avoids matrix estimation and assumes separability of the function components.

3. MATRIX-FREE TRIANGULAR UPDATES VIA PARTIAL MATRIX ESTIMATION

This section describes our first proposal for computing and applying the triangular updates for a sequence of linear systems in matrix-free environment. As mentioned above, in general, it is possible to obtain a system matrix by solving the *matrix estimation problem*, i.e. Problem 2.1. We will see that we can efficiently estimate only a part of a given matrix, that is, we will solve a *partial matrix estimation problem* [37, 39].

Using the notation from above, consider matrices A and A^+ from a sequence. If we need to compute a preconditioner directly from A^+, then a straightforward strategy is to estimate A^+ entirely. When the sparsity patterns of A^+ and A are the same, we can use the same graph G to find, let us say, p color groups for both matrices (note that we typically use only approximate algorithms for graph coloring since the related decision problem is NP-complete [40]), and the graph coloring algorithm does not need to be rerun to estimate A^+ if we have estimated A. In this way, we need p matvecs for each estimation. If the matrix patterns in the sequence differ too much, we may need to run the graph coloring algorithm for A^+ as well, but its running time is typically smaller than the time needed for the matvecs. It was demonstrated in [39] that for A^+ we can use the results of the graph coloring algorithm for a matrix with a 'slightly different' sparsity pattern.

In order to use the triangular updates described above, we only have to estimate, in addition to A which was estimated earlier, the upper or the lower triangular part of A^+. This leads to a special partial matrix estimation problem. Without loss of generality, consider estimation of the lower triangular part $\mathrm{tril}(A^+)$ of A^+. We will formulate this problem as a standard graph coloring problem (called 1-distance graph coloring problem; the problem can also be formulated differently using a different vertex coloring paradigm) for a graph which is different from the intersection graph of A^+. The following theorem describes this graph.

Theorem 3.1
Consider the graph

$$G_T(B)=G(\mathrm{tril}(B)^{\mathrm{T}}\mathrm{tril}(B))\cup G_K,$$

where $G(\mathrm{tril}(B)^{\mathrm{T}}\mathrm{tril}(B))=(V,E)$ is the intersection graph of the lower triangular part of the matrix B and G_K is defined as

$$G_K=\bigcup_{i=1}^{n} G_i, \quad G_i=(V_i,E_i)=(V,\{\{k,j\}|\ b_{ik}\neq 0\wedge b_{ij}\neq 0\wedge k\leqslant i<j\}).$$

DOI: 10.1002/nla

If the graph $G_T(B)$ can be colored by p colors, then the entries of the lower triangular part tril(B) of B can be computed by p matvecs of B with vectors $d_1, \ldots, d_p$ such that for each nonzero entry b_{ij} of tril(B) there is a vector d_k, $1 \leqslant k \leqslant p$, satisfying $(Bd_k)_i = b_{ij}(d_k)_j$.

Proof

First note that the theorem gives necessary conditions to solve a modified Problem 2.1 in which we have to estimate only the entries of tril(B) via matvecs with B. The intersection graph $G(\text{tril}(B)^{\mathrm{T}}\text{tril}(B))$ prohibits the vectors $d_1, \ldots, d_p$ to contain in any component a sum of two or more nonzero entries from tril(B). The graph G_K then prohibits to have in any of these vectors a component which would add a nonzero entry of tril(B) with one or more nonzero entries of the strict upper triangular part of the adjacency graph $G(B^{\mathrm{T}}B)$ of B. Note that the role of the index i in the definition of G_K restricts the adjacency of the entries b_{ik} and b_{ij} just to the cases when the former is from tril(B) and the latter from the strict upper triangular part of B.

Assume now that $G_T(B)$ was colored by p colors. Define the vectors as usual in matrix estimation problems, that is d_k, $1 \leqslant k \leqslant p$, such that

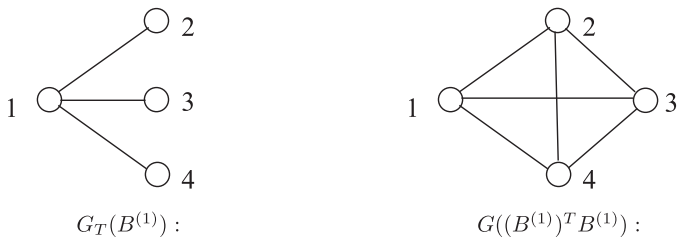
$$(d_k)_j = \begin{cases} 1 & \text{if the vertex } j \text{ has the color } k, \\ 0 & \text{otherwise.} \end{cases}$$

Consider a nonzero entry b_{ij} of tril(B). There are edges $\{i, l\}$ in $G_T(B)$ for each $1 \leqslant l \leqslant n$ such that b_{il} is nonzero and their existence is a sufficient condition for separate computability of the entries of tril(B). Note that this would not necessarily hold for a nonzero entry of B outside tril(B). It also need not to be the case if G_K would be missing as we explained in this proof above. Then we have $(Bd_k)_i = b_{ij}$ for some k and we have the result. □

Note that the graph $G_T(B)$ contains only a subset of edges of the adjacency graph $G(B^{\mathrm{T}}B)$, which should be considered to solve Problem 2.1. Consequently, in order to estimate only a triangular part of A^+, we may need a *smaller* number of matvecs than in the case of estimation of the whole A^+ (and subsequent extraction of the desired triangular part). We will show this in the following example.

Example 3.1

$$B^{(1)} = \begin{pmatrix} * & * & * & * \\ * & * & & \\ * & & * & \\ * & & & * \end{pmatrix},$$



$G_T(B^{(1)})$: $G((B^{(1)})^T B^{(1)})$:

DOI: 10.1002/nla

The graph $G_T(B^{(1)})$ for the estimation of the lower triangular part can be colored with two colors, but the graph $G((B^{(1)})^{\mathrm{T}}B^{(1)})$ for estimating the whole matrix needs all four colors. Hence in general, the lower triangular part of an n-dimensional matrix with the sparsity pattern of $B^{(1)}$ can always be estimated with two colors, whereas estimating the whole matrix will require n colors.

This extreme situation will not arise often in practice, but nevertheless one can save an important amount of matvecs by restricting estimation to one triangular part. In order to enhance this effect, we also propose to perform a prefiltration that decreases the size of G_T. The prefiltration is based on the sparsity pattern of the reference matrix. We summarize the two crucial points of the approach we propose, i.e. partial estimation and prefiltration, in the algorithm below. The final matrix-free preconditioned iterative method with the updates to solve a sequence of linear systems needs to estimate the reference matrix as well as the triangular parts of the remaining matrices, so that they could be used in the updates. The following algorithm describes how these two tasks can be performed.

Algorithm 3.1
PARTIAL MATRIX ESTIMATION FOR TRIANGULAR PRECONDITIONER UPDATES. ***Input:*** *Matrix sequence* $A^{(0)}, A^{(1)}, \ldots, A^{(n)}$ *and the sparsity pattern* $\mathcal{S}(A^{(0)})$.

1. **Estimation.** *Estimate* $A^{(0)}$ *using* $\mathcal{S}(A^{(0)})$.
2. **Initial factorization.** *Factorize* $A^{(0)}$ *such that* $A^{(0)} \approx LDU$.
3. **Sparsification.** *Filtrate* $A^{(0)}$ *to get* $\overline{A^{(0)}}$ *and its sparsity pattern* $\mathcal{S}(\overline{A^{(0)}})$.
4. ***for*** $i = 0, \ldots, n$
   Estimate the lower triangular part $\mathrm{tril}(A^{(i)})$ *of* $A^{(i)}$ *using the coloring of* $G_T(\overline{A^{(0)}})$ *and matrix-vector products with* $A^{(i)}$. *Then for* $i \geqslant 1$, *the lower triangular part of the difference matrix*, $\mathrm{tril}(A^{(0)}) - \mathrm{tril}(A^{(i)})$, *is used for the updates of the form* (2).
   end for

Note that the lower triangular part $\mathrm{tril}(A^{(0)})$ of $A^{(0)}$ is estimated twice. First, it is estimated as part of the whole matrix with the original sparsity pattern. Second, in Step 4 of Algorithm 3.1, the filtrated pattern $\mathcal{S}(\overline{A^{(0)}})$ is used. Based on our experiments, the updates use just the latter quantity, that is, the updates use the lower triangular parts $\mathrm{tril}(B^{(i)}) = \mathrm{tril}(A^{(0)}) - \mathrm{tril}(A^{(i)})$, for $i = 1, \ldots, n$, which were all computed with the *filtrated and approximate* sparsity pattern $\mathcal{S}(\overline{A^{(0)}})$. Since the estimation adds some error to the computed matrix entries, it is important to distribute this error in all the approximate matrices in the same way. This explains our choice and the fact that the loop in Step 4 starts from 0. As for the sequential graph coloring heuristic, it tries to balance the error among the groups of columns of the same color as proposed in [39], see also [41]. Table II displays the costs and memory for one step of an iterative method preconditioned through recomputation and update, respectively, in matrix-free environment where updating is based on the approach from this section. We denote matrix estimations by $\mathrm{est}(\cdot)$. We see that in the initialization we save not only all factorization costs, but also estimation costs by solving a partial estimation problem.

An interesting aspect of the estimation of a triangular part of a matrix is that the number of colors depends on the matrix reordering since the graph construction depends on it. This is in

DOI: 10.1002/nla

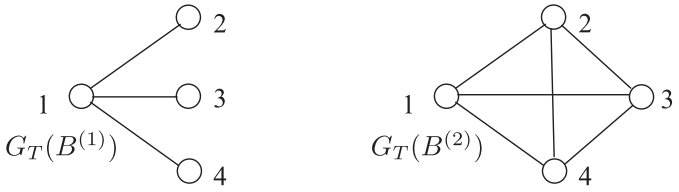
Table II. Cost comparison with matrix-free updates based on Algorithm 3.1.

Type	Initialization	Solve step	Memory
Recomp	est(A^+), $A^+ \approx L^+D^+U^+$	Solves with L^+D^+, U^+	L^+D^+, U^+
Update	est(tril(A^+))	Solves with LD, U, tril(B)	tril(A^+), tril(A), LD, U

contrast with the standard matrix estimation for the whole matrix explained in the previous section. It is easy to see this from Example 3.2, where we show the arrow matrix $B^{(1)}$ from Example 3.1 and its reordering $B^{(2)}$ together with the corresponding graphs $G_T(B^{(1)})$ and $G_T(B^{(2)})$. We have chosen this simple example to show the contrast between the Cuthill–McKee (CM) and the reverse Cuthill–McKee reorderings (RCM) [42]. While $G_T(B^{(1)})$ reminds a part of the recursive structure which we get from the CM algorithm, $G_T(B^{(2)})$ shows a typical structure after reversing the sequence. The following theorem shows that sometimes we can enhance our chances to decrease the number of matvecs needed for the estimation of a triangular part of the matrix by an appropriate ordering of the matrix. Counterintuitively, the reverse CM reordering may not be generally recommended.

Example 3.2

$$B^{(1)} = \begin{pmatrix} * & * & * & * \\ * & * & & \\ * & & * & \\ * & & & * \end{pmatrix} \qquad B^{(2)} = \begin{pmatrix} * & & & * \\ & * & & * \\ & & * & * \\ * & * & * & * \end{pmatrix}$$



Theorem 3.2
Assume that the irreducible matrix B with symmetric sparsity pattern was reordered by the CM reordering. Further assume that the following condition applies: if $b_{ij} \neq 0$ for some i, j, $1 \leqslant j \leqslant i \leqslant n$, then $b_{lj} \neq 0$ for all l, $j \leqslant l \leqslant i$ (envelope assumption). Denote by $\hat{B}$ the matrix which we obtain from B by reversing the order of rows and columns with respect to B, that is, $\hat{B}$ corresponds to the original matrix reordered by the related RCM reordering. Then the chromatic number of $G_T(B)$ is not larger than the chromatic number $G_T(\hat{B})$.

Proof
We will use induction on i. Let us first define $f_i = \min\{j | b_{ij} \neq 0\}$ for $1 \leqslant i \leqslant n$. If B is reordered by the CM reordering then we know that $f_i \leqslant f_j$ if $1 \leqslant i \leqslant j \leqslant n$ (monotone envelope property) and $f_i < i$ for $1 < i \leqslant n$ [43].

DOI: 10.1002/nla

$$\begin{pmatrix} * & * & * & & & \\ * & * & & * & & \\ * & & * & & * & * \\ & * & & * & & * \\ & & * & & * & * \\ & & * & * & * & * \end{pmatrix}$$

Figure 1. Matrix B after the Cuthill–McKee reordering for which its graph $G_T(B)$ needs more colors than the graph $G_T(\hat{B})$, where we get $\hat{B}$ from B after the symmetric reversal of its columns and rows.

The assertion is trivially valid for $i=1$. Consider $i>1$. Assume that we border the matrix of dimension $i-1$ by a row i from the bottom and a column i from the right. Let j be the minimum index such that $b_{ij}\neq 0$. The nonzero entries in the ith row induce a complete subgraph in $G_T(B)$. Because of the envelope assumption, this complete subgraph must be in $G_T(\hat{B})$ as well since the nonzeros b_{lj} for $j\leqslant l\leqslant i$ induce a clique. Consequently, $G_T(\hat{B})$ has all edges from $G_T(B)$ and its chromatic number is not smaller than the chromatic number of $G_T(B)$. □

Note that CM/RCM reorderings are often used to preprocess a matrix of linear systems solved by preconditioned iterative methods. One motivation in the nonsymmetric case is that such reorderings may be very beneficial for the stability of the incomplete decomposition [44]. Further note that Theorem 3.2 is not valid without the envelope assumption. Figure 1 shows an example of a matrix B after the CM reordering. The chromatic number of $G_T(B)$ is five. B reordered by the related RCM needs only four colors. Despite this artificial counterexample, Theorem 3.2 points out that the CM reordering may generally be preferable when estimating the lower triangular part of a matrix.

4. MATRIX-FREE UPDATES VIA IMPLICIT BACK- OR FORWARD SOLVES

This section describes an approach for applying the triangular preconditioner updates (2) in matrix-free environment which is principally different from the previous one. Whereas in the previous section we focused on efficient estimation of matrix entries, here we will avoid estimation as much as possible. We will show that we can apply the updates without knowledge and storage of the entries of the difference matrix B. This approach is therefore closer to the philosophy of matrix-free environment than the approach of the latest section. The price we pay for the reduced storage costs are forward or backward solves based on function evaluations while applying the updated preconditioner. The resulting increase in computational costs depends on the price of the function evaluation and on the degree of separability of the components of the function as defined in Section 2.

Assume for the moment that the triangular part of the matrix A and its incomplete LU decomposition are available explicitly (for simplicity, we hide the diagonal factor D of the decomposition in L). In practice, these quantities are computed for the reference system of the sequence. Let the current matrix A^+ be given implicitly in the form of its action on vectors, expressed by the

DOI: 10.1002/nla

function evaluation $\mathscr{F}^+(\cdot)$, where $\mathscr{F}^+:\mathbb{R}^n\to\mathbb{R}^n$ and let $\mathscr{F}_i^+$ be the ith component of $\mathscr{F}^+$. The strategy to apply the updated preconditioner

$$(L-\mathrm{tril}(B))U \tag{3}$$

which we propose when the function components are separable is summarized in Algorithm 4.1.

Algorithm 4.1
APPLICATION OF TRIANGULAR PRECONDITIONER UPDATES WITH MIXED EXPLICIT-IMPLICIT SOLVES. ***Input***: *Explicitly stored matrices* L, U *and* $\mathrm{tril}(A)$ *and the function components of* $\mathscr{F}^+$ *which represent* A^+ *implicitly.*

1. **Initialization.** *Find the main diagonal* $\{a_{11}^+,\ldots,a_{nn}^+\}$ *of* A^+ *before running the iterative method. It can be found by computing*

$$a_{ii}^+=\mathscr{F}_i^+(e_i),\quad 1\leqslant i\leqslant n.$$

2. **Forward solve in each iteration.** *Use the following mixed explicit–implicit strategy: Split the lower triangular matrix of* (3) *as* $L-\mathrm{tril}(B)=E+\mathrm{tril}(A^+)$. *That is,* $E\equiv L-\mathrm{tril}(A)$ *is stored explicitly, and the implicit part* $\mathrm{tril}(A^+)$ *contains entries of the new system matrix. We then have to solve triangular systems of the form*

$$(E+\mathrm{tril}(A^+))z=y,$$

which yields the forward solve loop

$$z_i=\frac{y_i-\sum_{j<i}e_{ij}z_j-\sum_{j<i}a_{ij}^+z_j}{e_{ii}+a_{ii}^+},\quad i=1,2,\ldots,n. \tag{4}$$

Note that the values e_{ii} *and* a_{ii}^+ *in the denominator are known. In the numerator of* (4), *the first sum can be computed explicitly and the second sum can be computed with the function component evaluation*

$$\mathscr{F}_i^+((z_1,\ldots,z_{i-1},0,\ldots,0)^{\mathrm{T}})\approx e_i^{\mathrm{T}}A^+(z_1,\ldots,z_{i-1},0,\ldots,0)^{\mathrm{T}}=\sum_{j<i}a_{ij}^+z_j. \tag{5}$$

3. **Backward solve in each iteration.** *This is a trivial step since the matrix* U *in* (3) *has been stored explicitly.*

The costs to find the main diagonal in Step 1 (initialization) correspond approximately to the costs of one full function evaluation if the function components are well separable. Step 1 needs to be performed only once, before the preconditioned Krylov subspace method is started. Note that the diagonal of $B=A-A^+$ is known in advance in some applications. In particular, if the diagonal does not change, $\mathrm{diag}(B)$ is the zero matrix. In Step 2, the whole forward-solve loop requires n partial evaluations (5). In total, this gives approximately the cost of one additional full function evaluation per solve step of the preconditioned iterative method.

We assumed above that the triangular part $\mathrm{tril}(A)$ of the reference matrix is stored explicitly. We mention that if the estimation of A has not been efficient or if the sparsity patterns of $\mathrm{tril}(A)$ and L differ so much that the storage costs would grow unacceptably, one may also use the function components of $\mathscr{F}$, corresponding to A, to replace operations with $\mathrm{tril}(A)$ in the forward solve.

DOI: 10.1002/nla

Table III. Costs comparison with matrix-free updates based on Algorithm 4.1.

Type	Initialization	Solve step	Memory
Recomp	est(A^+), $A^+ \approx L^+U^+$	Solves with L^+, U^+	L^+, U^+
Update	est(diag(A^+))	Solves with L, U, eval($\mathscr{F}$, $\mathscr{F}^+$)	L, U

Then tril(A) does not need to be stored and, formally, the explicit part E of the forward solve consists of L only. Then in (4) there are three sums of which the last two are computed implicitly:

$$z_i = \frac{y_i - \sum_{j<i} e_{ij} z_j - \sum_{j<i} a_{ij} z_j - \sum_{j<i} a_{ij}^+ z_j}{e_{ii} + a_{ii} + a_{ii}^+}, \quad i = 1, 2, \ldots, n, \tag{6}$$

where a_{ij} represents entries of the reference matrix A. The sum $\sum_{j<i} a_{ij} z_j$ is computed as

$$\mathscr{F}_i((z_1, \ldots, z_{i-1}, 0, \ldots, 0)^{\mathrm{T}}).$$

The whole forward solve would then cost about two full function evaluations in total.

Let us compare the approach of this section with the strategy that recomputes the preconditioner for each system of a given sequence. With updates applied according to Algorithm 4.1, only the main diagonal of A^+ needs to be estimated (in the initialization). If we would recompute the preconditioner, on the other hand, we would have to estimate the whole matrix A^+, and, mainly, to compute the incomplete factorization. However, *application* of the update using Algorithm 4.1 could be more expensive than applying a new factorization if we would need a similar number of iterations since at least an extra full function evaluation in each forward solve is needed. Note that with the strategy of this section the memory costs for updating can be kept as low as they are for recomputing. The previous observations are displayed schematically in Table III for the case where forward solves are performed according to (6). We denote function evaluations by eval($\cdot$).

5. NUMERICAL EXPERIMENTS

This section is devoted to numerical experiments illustrating the techniques from Sections 3 and 4. The main focus is not to show that the considered preconditioner updates can be very beneficial as compared with freezing or recomputing preconditioners; this has been shown elsewhere [32–34]. Here we present experiments with the new algorithms proposed for matrix-free environment. We focus on illustrating the aspects of the proposed matrix-free implementation techniques and, in addition, the experiments show that updating is a robust alternative to freezing or recomputation in the matrix-free environment. We attempted to use a variety of ILU decompositions and we performed tests with GMRES as well as with BiCGSTAB. In all experiments we consider minimization of functions with Newton-type methods and we use, to avoid storing Jacobians, the standard difference approximation of the Jacobian of the function F that is to be minimized. More precisely, a matvec with the Jacobian, Av, is replaced by

$$\mathscr{F}(v) \equiv \frac{F(x + \varepsilon \cdot v/\|v\|) - F(x)}{\varepsilon}, \tag{7}$$

for some small $\varepsilon > 0$, where x is the point (vector) at which the Jacobian is approximated.

DOI: 10.1002/nla

We will consider two test problems. The first problem results from a Newton-type method with a flexible stopping criterion for the linear system solution. Here the preconditioners with the triangular updates were fully embedded into the nonlinear solver. The next problem considers, on the other hand, a fixed sequence of linear systems generated from a nonlinear solver. All experiments were implemented in Fortran 95 on Intel Pentium-based machines.

5.1. Test problem 1

In this first example we minimize a function with easily separable components resulting from a two-dimensional nonlinear convection–diffusion model problem with finite difference discretization. The convection–diffusion model problem has the form (see, e.g. [1])

$$-\Delta u + Cu\left(\frac{\partial u}{\partial x}+\frac{\partial u}{\partial y}\right)=f(x,y), \quad f(x,y)=2000x(1-x)y(1-y), \tag{8}$$

where $C>0$ is the Reynolds number, and it is discretized on the unit square. The standard five-point discretization stencil (central differences) results in a function F to minimize with components of the simple form

$$F_k(x)\equiv\frac{4x_k-x_{k-1}-x_{k+1}-x_{k+N}-x_{k-N}}{h^2}+Cx_k\frac{x_{k+1}-x_{k-1}+x_{k+N}-x_{k-N}}{h}-f_k,$$

for $1\leqslant k\leqslant n$, where f_k is the discretization of f and N is the number of inner nodes. To solve $F(x)=0$ we use an inexact Newton–Krylov method where the Krylov subspace method is BiCGSTAB and the stopping criterion of the iterative method is chosen adaptively. Newton's method is combined with a line-search technique for global convergence; the used method is described in detail on [45, p. 215], see method DNS. The final matrix-free solver was embedded into the UFO-software [46] for nonlinear problems. The initial approximation is the discretization of $u_0(x,y)=0$.

The preconditioner we use in the experiments is ILUT, see [47], and for our implementation we used Saad's ILUT code. We considered changes in the number of additional nonzeros allowed in the factorization, ranging from ILUT(tol, 0) $\equiv$ ILU(0) with the same sparsity pattern as the system matrix, to ILUT(tol, 5). The drop tolerance was always tol $=0.01$.

The ILUT factorizations are computed from the estimations of Jacobians obtained by running a graph coloring algorithm and performing matvecs corresponding to the obtained color groups. The graph coloring algorithm (which is based on a simple heuristic) yields between 5 and 7 colors; hence, the matrix is estimated with about 5–7 matvecs calculated through means of function evaluations of the form (7). When, for the updates, we run Algorithm 3.1 and estimate only one triangular part of the system matrices A^+, then this always yields 5 colors and thus it will require 5 matvecs. In other words, the graph $G_T(A^+)$ of Theorem 3.1 has approximately as many edges as the intersection graph $G((A^+)^TA^+)$ of the whole matrix in this example (in the other test problem the situation is different). This is due to the structure of the system matrices and to the fact that the filtration in Step 3 of Algorithm 3.1 does not sparsify the initial matrix (which is just a Laplacian). Of course, with the update strategy of Algorithm 4.1, the triangular parts of A^+ need not be computed at all.

In Figures 2, 3 and 4 we display results for several types of ILUT and different grid sizes. More precisely, Figures 2, 3 and 4 present experiments with the dimensions 62 500, 96 100 and 204 100, respectively, and for each dimension preconditioning with ILUT(0.01, f) for the fill parameters $f=0,1,\ldots,5$ is tested (see the x-axes of the graphs). Each figure contains two graphs where

DOI: 10.1002/nla

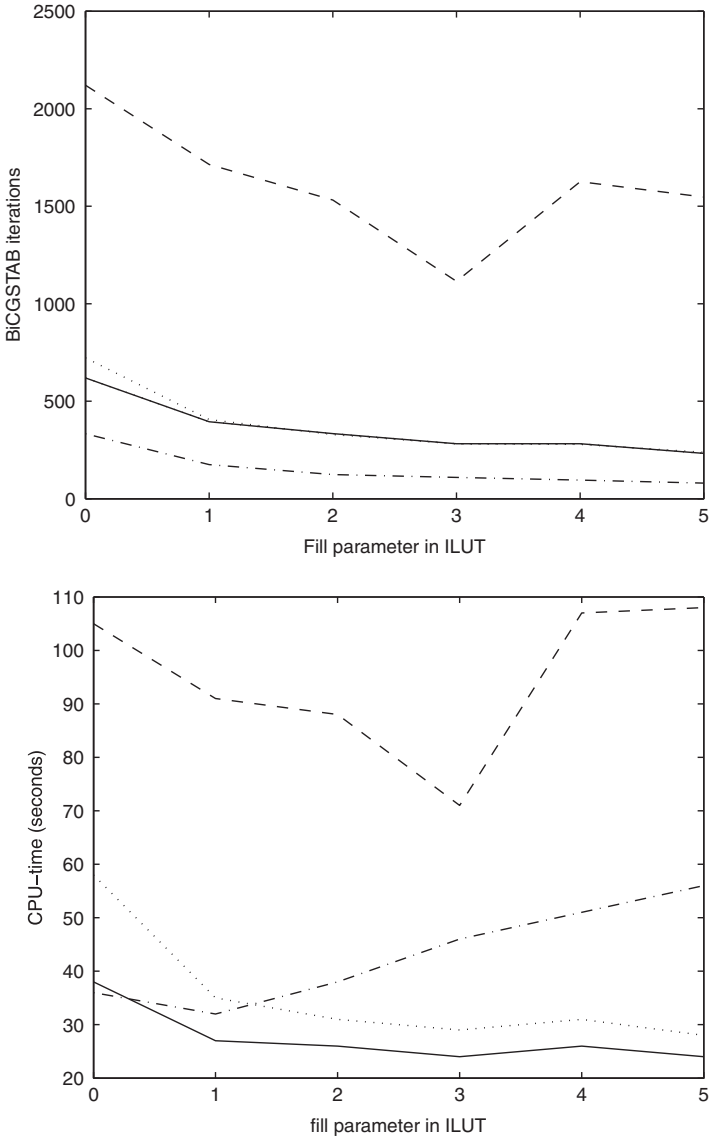


Figure 2. BiCGStab iterations and CPU-times to solve problem (8) with $C = 500$ on a 250×250 grid (dimension 62 500) with varying sizes of ILUT-factorizations (depending on the fill parameter) for freezing (dashed lines), recomputing (dash-dotted lines), updating with Algorithm 3.1 (solid lines) and updating with Algorithm 4.1 (dotted lines).

the first graph displays the number of BiCGSTAB iterations needed to reduce $\|F(x)\|$ to the value 10^{-15} and the second graph gives the needed CPU-time. The Reynolds number is chosen as $C = 500$, which yields sequences of about 10–12 linear systems. With this Reynolds number, ILU(0) is in general not the most efficient preconditioner; preconditioners with a number of nonzeros clearly larger than the system matrix yield less BiCGSTAB iterations (in this series of experiments, ILUT$(0.01, 5)$ has about three times as many nonzeros as the system matrix). The graphs compare

DOI: 10.1002/nla

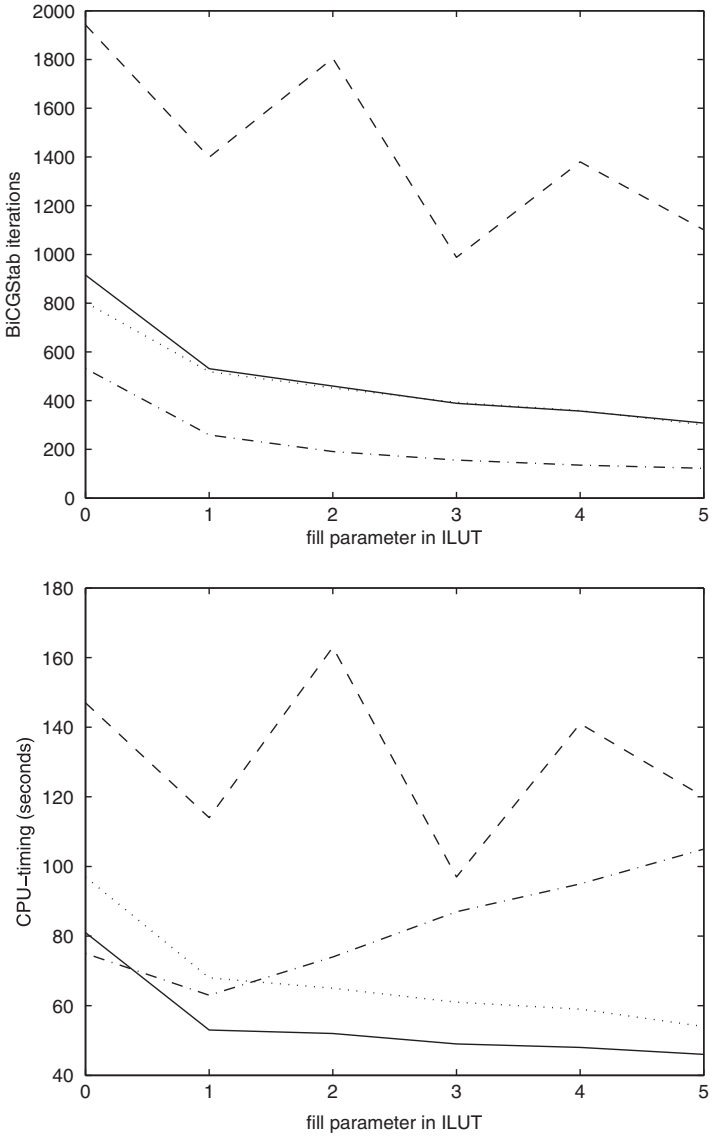


Figure 3. BiCGStab iterations and CPU times to solve problem (8) with $C = 500$ on a 310×310 grid (dimension 96 100) with varying sizes of ILUT-factorizations (depending on the fill parameter) for freezing (dashed lines), recomputing (dash-dotted lines), updating with Algorithm 3.1 (solid lines) and updating with Algorithm 4.1 (dotted lines).

four ways to precondition the sequences of linear systems: Dashed lines represent freezing of the preconditioner computed for the initial linear system, dash-dotted lines represent preconditioner recomputation for every linear system of the sequence, solid lines represent matrix-free updating based on (2) applied with Algorithm 3.1 and dotted lines represent matrix-free updating based on (2) applied with Algorithm 4.1. The choice between the two update types in (2) was made adaptively according to the triangular part of $B^1 = A^0 - A^1$ with the entries largest in magnitude

DOI: 10.1002/nla

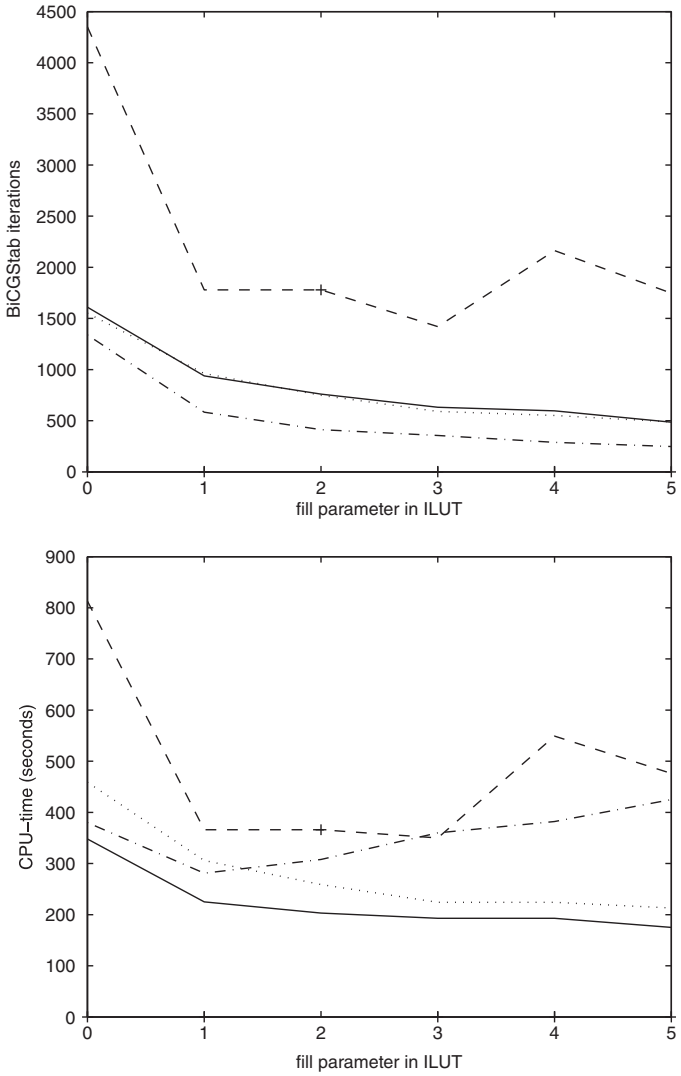


Figure 4. BiCGStab iterations and CPU-times to solve problem (8) with $C = 500$ on a 490×490 grid (dimension 240 100) with varying sizes of ILUT-factorizations (depending on the fill parameter) for freezing (dashed lines), recomputing (dash-dotted lines), updating with Algorithm 3.1 (solid lines) and updating with Algorithm 4.1 (dotted lines).

(as proposed in [32]). This was always the upper triangular part, hence we always updated this triangular part.

The first observation which we get from the figures is that the total number of BiCGSTAB iterations needed to solve (8) is lowest when we recompute from scratch. However, updating based on (2) follows the curves for recomputing at close distance. As one would expect, the difference between using Algorithm 3.1 and Algorithm 4.1 for updating is marginal because they are just different implementations (i.e. based on different matrix-free computation techniques) of the same updated preconditioner. The number of iterations needed with the frozen preconditioner deteriorates

DOI: 10.1002/nla

soon during the solution of the sequence and is rather unpredictable. In particular, there is no clear dependence on the fill parameter f of the ILUT$(0.01, f)$ factorization. The plus '+' in Figure 4 indicates that the corresponding frozen ILUT factorization was too weak to solve the sequence arising from the nonlinear problem (8) at all.

The situation is quite different when we consider CPU-time. The repeated computation of the ILUT factorization is relatively expensive in this example of a matrix-free implementation and therefore the recomputation strategy is in general clearly less time efficient than updating. An exception is given by the case of ILU(0) factorizations, which are, of course, cheap to (re)compute. For example, for the first series of tests with dimension $n = 62\,500$ the average time to compute ILU(0) is 1.5 s and the average computation times of ILUT$(0.01, 1)$, ILUT$(0.01, 2)$, ILUT$(0.01, 3)$, ILUT$(0.01, 4)$ and ILUT$(0.01, 5)$ are, respectively, 1.8, 2.6, 3.3, 3.7 and 4.3 s. Seen the length of the linear system sequence, this represents a considerable part of the total solution time when we use the recomputation strategy. For larger factorizations we observe that recomputing takes longer for denser factorizations, whereas updating is the *faster* the denser is the factorization. Algorithm 3.1 is more time efficient than Algorithm 4.1; this is what one expects as Algorithm 4.1 requires about one additional function evaluation per backward solve with the updated preconditioner, hence two more function evaluations per BiCGSTAB iteration (in BiCGSTAB the preconditioner is applied twice in every iteration). On the other hand, Algorithm 4.1 has the advantage that no triangular parts of the current system matrices need to be stored (or estimated). If we would perform the forward solves according to (6), then we would not even need to store the lower triangular part of the reference system matrix, but the whole computation would be slower than when using Algorithm 4.1. The freezing strategy performs in general worst of all. This is explained by the deteriorating number of BiCGSTAB iterations, but note that in Figure 4 the timing of freezing and recomputing is the same though freezing needs much more BiCGSTAB iterations for convergence (this shows the high costs of recomputing).

5.2. Test problem 2

The second set of experiments is devoted to solving a sequence of linear problems arising during the computation of a constitutive model from structural mechanics provided by Karsten Quint. More precisely, a small strain metal viscoplasticity model was developed for a rectangular plate of length 100, width 21.2 and height 9.62 cm with a hole in the middle. The discretization used 1 350 quadratic elements in most of the domain with a somewhat finer grid in the center. When applying the Multilevel-Newton algorithm, every time step contains an inner loop that requires the solution of nonlinear systems, which in turn leads to a sequence of linear systems. For more details on the parameters of the material and of the Multilevel-Newton algorithm which were used, we refer to the description of the first application in [48]. We consider here a sequence of linear systems from a randomly chosen time-step in the middle of the simulation process. This sequence consists of 8 linear systems of dimension 4 936 with matrices containing about 315 000 nonzeros. To solve the sequence, we use the GMRES(40) method preconditioned by ILUT. The (fixed) stopping criterion for GMRES is relative reduction of the residual norm below the level 10^{-6}. We again present results of several experiments differing in parameters provided to the preconditioner. In particular, we would like to show that the matrix-free updating strategies are successful over large variations in the preconditioner density.

Tables IV–VIII present the results in terms of number of GMRES iterations and needed matrix-vector multiplications (that is, the function evaluations (7)). We compare again the four

DOI: 10.1002/nla

Table IV. Number of iterations and function evaluations for solving preconditioned linear systems from the structural mechanics problem with ILUT(0.01, 5).

	Recomp		Freeze		Update (Algorithm 3.1)		Update (Algorithm 4.1)	
Matrix	its	est. fevals	its	est. fevals	its	est. fevals	its	est. fevals
ILUT(0.01, 5), Psize $\approx$ 260 000								
$A^{(0)}$	343	89	343	89	343	89+25	343	89
$A^{(1)}$	172	89	623	0	237	25	237	0
$A^{(2)}$	201	89	694	0	298	25	298	0
$A^{(3)}$	294	89	723	0	285	25	285	0
$A^{(4)}$	298	89	799	0	334	25	334	0
$A^{(5)}$	386	89	708	0	320	25	320	0
$A^{(6)}$	348	89	714	0	318	25	318	0
$A^{(7)}$	317	89	717	0	318	25	318	0
overall fevals		3071		5410		2742		4652

Table V. Number of iterations and function evaluations for solving preconditioned linear systems from the structural mechanics problem with ILUT(0.001, 20).

	Recomp		Freeze		Update (Algorithm 3.1)		Update (Algorithm 4.1)	
Matrix	its	est. fevals	its	est. fevals	its	est. fevals	its	est. fevals
ILUT(0.001, 20), Psize $\approx$ 404 000								
$A^{(0)}$	187	89	187	89	187	89+25	187	89
$A^{(1)}$	89	89	393	0	146	25	146	0
$A^{(2)}$	126	89	448	0	182	25	182	0
$A^{(3)}$	221	89	480	0	184	25	184	0
$A^{(4)}$	234	89	513	0	190	25	190	0
$A^{(5)}$	193	89	487	0	196	25	196	0
$A^{(6)}$	178	89	521	0	196	25	196	0
$A^{(7)}$	246	89	521	0	196	25	196	0
overall fevals		2186		3639		1766		2856

computational strategies: preconditioner recomputation by matrix estimation for each system (Recomp), preconditioner computation only for the reference matrix (Freeze), and preconditioning with the triangular updates based on Algorithm 3.1 and on Algorithm 4.1. Let us remind that Algorithm 3.1 evaluates in its loop also a less accurate approximation of the triangular part of $A^{(0)}$ using the filtrated pattern $\mathcal{S}(\overline{A^{(0)}})$, which we use to compute the updates, although we have a more accurate $A^{(0)}$ available. $A^{(0)}$ was filtrated in Algorithm 3.1 such that all the entries with magnitude smaller than half of the magnitude of the largest entry in their rows were dropped.

The average number of nonzeros of the factorizations is denoted with 'Psize'. The column 'est. fevals' gives the number of function evaluations (fevals) needed for matrix estimation and 'overall fevals' present the total number of fevals needed to solve the sequence. The two 'est. fevals' numbers for $A^{(0)}$ in the column 'Update (Algorithm 3.1)' correspond to its estimation with full and filtrated pattern. The other 'est. fevals' numbers in this column give the number

DOI: 10.1002/nla

Table VI. Number of iterations and function evaluations for solving preconditioned linear systems from the structural mechanics problem with ILUT($10^{-4}, 30$).

	Recomp		Freeze		Updated (Algorithm 3.1)		Updated (Algorithm 4.1)	
Matrix	its	est. fevals	its	est. fevals	its	est. fevals	its	est. fevals
ILUT($10^{-4}, 30$), Psize$\approx$550 000								
$A^{(0)}$	85	89	85	89	85	89+25	85	89
$A^{(1)}$	59	89	233	0	78	25	78	0
$A^{(2)}$	72	89	313	0	84	25	84	0
$A^{(3)}$	78	89	344	0	85	25	85	0
$A^{(4)}$	78	89	289	0	108	25	108	0
$A^{(5)}$	78	89	289	0	108	25	108	0
$A^{(6)}$	79	89	318	0	108	25	108	0
$A^{(7)}$	86	89	318	0	108	25	108	0
overall fevals		1 327		2 278		1 053		1 532

Table VII. Number of iterations and function evaluations for solving preconditioned linear systems from the structural mechanics problem with ILUT($10^{-5}, 50$).

	Recomp		Freeze		Updated (Algorithm 3.1)		Updated (Algorithm 4.1)	
Matrix	its	est. fevals	its	est. fevals	its	est. fevals	its	est. fevals
ILUT($10^{-5}, 50$), Psize$\approx$812 000								
$A^{(0)}$	65	89	65	89	65	89+25	65	89
$A^{(1)}$	31	89	128	0	52	25	52	0
$A^{(2)}$	35	89	163	0	45	25	45	0
$A^{(3)}$	35	89	237	0	45	25	45	0
$A^{(4)}$	37	89	167	0	52	25	52	0
$A^{(5)}$	38	89	169	0	51	25	51	0
$A^{(6)}$	37	89	168	0	51	25	51	0
$A^{(7)}$	50	89	168	0	51	25	51	0
overall fevals		1 040		1 354		701		848

of matvecs needed to estimate only the triangular part of the current system matrix. To compute 'overall fevals' we counted one function evaluation per GMRES iteration, as there is one matvec with the system matrix in every iteration of the GMRES method. In the 'Update (Algorithm 4.1)' strategy, however, every application of the updated preconditioner requires an additional function evaluation. Therefore we counted two function evaluations per GMRES iteration for this strategy.

The first fact which we observe is that the updating strategy works very well in terms of iteration counts and, in contrast with the previous test problem, it is from this point of view only slightly worse than recomputing. Consequently, it is clear that the updates are very often able to recover a lot of the information missing in the LU decomposition of the reference matrix. Second we observe that in terms of function evaluations, updating with Algorithm 3.1 is always the cheapest of all strategies. This is for an important part due to the difference between estimating the whole matrix and estimating only one triangular part.

DOI: 10.1002/nla

Table VIII. Number of iterations and function evaluations for solving preconditioned linear systems from the structural mechanics problem with ILUT($10^{-6}, 70$).

	Recomp		Freeze		Updated (Algorithm 3.1)		Updated (Algorithm 4.1)	
Matrix	its	est. fevals	its	est. fevals	its	est. fevals	its	est. fevals
ILUT($10^{-6}, 70$), Psize$\approx$950000								
$A^{(0)}$	32	89	32	89	32	89+25	32	89
$A^{(1)}$	21	89	78	0	54	25	54	0
$A^{(2)}$	28	89	88	0	38	25	38	0
$A^{(3)}$	24	89	101	0	39	25	39	0
$A^{(4)}$	26	89	92	0	38	25	38	0
$A^{(5)}$	26	89	87	0	38	25	38	0
$A^{(6)}$	26	89	86	0	38	25	38	0
$A^{(7)}$	28	89	86	0	38	25	38	0
overall fevals		923		739		604		687

Updating with Algorithm 4.1 requires less function evaluations than recomputing only for the densest factorizations in our series of experiments. It requires always more function evaluations than with Algorithm 3.1 (though the difference becomes smaller with more powerful initial factorizations). In addition, separate computation of the function components as needed in (5) is rather expensive due to the given finite volume implementation. This is the price we pay for the lower memory demands when using Algorithm 4.1. By using Algorithm 4.1 we save the storage of triangular parts of the size of about 160.000 nonzeros. Note that there may be applications where one cannot afford to store tril(B) or triu(B) in addition to the factors L and U at all and using the mixed explicit-implicit solves of Algorithm 4.1 would be the only feasible updating option. Both recomputation and updates are in general much better than the freezing strategy. An exception is given in Table VIII where the ILUT($10^{-6}, 70$) factorization gives such low GMRES iteration counts that the estimation and recomputation costs start to dominate; here freezing is cheaper than recomputing.

As this sequence is a fixed sequence extracted from a structural mechanics problem solver and the experiments were not embedded in the solver, we do not present CPU timings and use just the number of function evaluations to get an idea of the computational effort. Note however, that if recomputing gives numbers of GMRES iterations close to updating, then its total timing, including factorization times, must necessarily be much worse. Let us also remind the experimental dependence of timings and iteration counts presented for the triangular updates in [32] if we assume similar sizes of preconditioners used in the compared strategies.

6. CONCLUSIONS

We have presented theoretical results and numerical experiments related to matrix-free strategies for solving sequences of linear systems by preconditioned iterative methods. In particular, we introduced two new approaches to apply triangular updates for enhancing the linear solver of the sequences. The experiments in matrix-free environment seem to confirm that the proposed

DOI: 10.1002/nla

strategies are typically the best of all compared possibilities. Moreover, the updates can be easily embedded into matrix-free nonlinear solvers.

ACKNOWLEDGEMENTS

We would like to gratefully acknowledge the help of Stefan Hartmann, Reijo Kouhia and Karsten Quint who provided the test problems. We are also very much indebted to Ladislav Lukšan for helping us to embed the updated preconditioners into the UFO software.

This work was supported by the project No. IAA100300802 of the Grant Agency of the Academy of Sciences of the Czech Republic. The work of the first author was also supported by project number KJB100300703 of the Grant Agency of the Academy of Sciences of the Czech Republic. The work of the second author was also supported by the international collaboration support M100300902 of AS CR.

REFERENCES

1. Kelley CT. *Iterative Methods for Linear and Nonlinear Equations*. SIAM: Philadelphia, PA, 1995.
2. Kelley CT. *Solving Nonlinear Equations with Newton's Method*. Fundamentals of Algorithms. SIAM: Philadelphia, PA, 2003.
3. Brown PN, Saad Y. Hybrid Krylov methods for solving systems of nonlinear equations. *SIAM Journal on Scientific and Statistical Computing* 1990; **11**:450–481.
4. Mousseau VA, Knoll DA, Rider WJ. Physics-based preconditioning and the Newton–Krylov method for non-equilibrium radiation diffusion. *Journal of Computational Physics* 2000; **160**:743–765.
5. Keyes D. *Terascale Implicit Methods for Partial Differential Equations*. Contemporary Mathematics, vol. 306. AMS: Providence, RI, 2001; 29–84.
6. Reisner J, Mousseau VA, Wyszogrodzki AA, Knoll DA. An efficient physics-based preconditioner for the fully implicit solution of small-scale thermally driven atmospheric flows. *Journal of Computational Physics* 2003; **189**:30–44.
7. Knoll DA, Keyes D. Jacobian-free Newton–Krylov methods: a survey of approaches and applications. *Journal of Computational Physics* 2004; **193**:357–397.
8. Bernsen E, Dijkstra HA, Wubs FW. A method to reduce the spin-up time of ocean models. *Ocean Modelling* 2008; **20**:380–392.
9. Li X, Primeau F. A fast Newton–Krylov solver for seasonally varying global ocean biogeochemistry models. *Ocean Modelling* 2008; **23**:13–20.
10. Khatiwala S. Fast spin up of ocean biogeochemical models using matrix-free Newton–Krylov. *Ocean Modelling* 2008; **23**:121–129.
11. Chan T, Jackson K. Nonlinearly preconditioned Krylov subspace methods for discrete Newton algorithms. *SIAM Journal on Scientific and Statistical Computing* 1984; **5**(3):533–542.
12. Luo H, Baum JD, Löhner R. A fast, matrix-free implicit method for compressible flows on unstructured grids. *Journal of Computational Physics* 1998; **146**(2):664–690.
13. Choquet R. A matrix-free preconditioner applied to CFD. *Technical Report No. 940*, INRIA, France, 1995.
14. Chen Y, Shen C. A Jacobian-free Newton-GMRES(m) method with adaptive preconditioner and its application for power flow calculations. *IEEE Transactions on Power Systems* 2006; **21**(3):1096–1103.
15. Morales JL, Nocedal J. Automatic preconditioning by limited memory quasi-Newton updating. *SIAM Journal on Optimization* 2000; **10**(4):1079–1096 (electronic).
16. Bergamaschi L, Bru R, Martínez A, Putti M. Quasi-Newton preconditioners for the inexact Newton method. *Electronic Transactions on Numerical Analysis* 2006; **23**:63–74.
17. Nocedal J. Updating quasi-Newton matrices with limited storage. *Mathematical Computations* 1980; **35**(151): 773–782.
18. Serra Capizzano S, Tablino Possio C. High-order finite difference schemes and Toeplitz based preconditioners for elliptic problems. *Electronic Transactions on Numerical Analysis* 2000; **11**:55–84 (electronic).
19. Serra Capizzano S, Tablino Possio C. Preconditioning strategies for 2D finite difference matrix sequences. *Electronic Transactions on Numerical Analysis* 2003; **16**:1–29 (electronic).

DOI: 10.1002/nla

20. Bertaccini D, Golub GH, Serra Capizzano S, Tablino Possio C. Preconditioned HSS methods for the solution of non-Hermitian positive definite linear systems and applications to the discrete convection–diffusion equation. *Numerische Mathematik* 2005; **99**(3):441–484.
21. Kharchenko SA, Yeremin AY. Eigenvalue translation based preconditioners for the GMRES(k) method. *Numerical Linear Algebra with Applications* 1995; **2**(1):51–77.
22. Baglama J, Calvetti D, Golub GH, Reichel L. Adaptively preconditioned GMRES algorithms. *SIAM Journal on Scientific Computing* 1998; **20**:243–269.
23. Erhel J, Burrage K, Pohl B. Restarted GMRES preconditioned by deflation. *Journal of Computational and Applied Mathematics* 1996; **69**(2):303–318.
24. Morgan RB. A restarted GMRES method augmented with eigenvectors. *SIAM Journal on Matrix Analysis and Applications* 1995; **16**(4):1154–1171.
25. Saad Y, Yeung M, Erhel J, Guyomarc'h F. A deflated version of the conjugate gradient algorithm. *SIAM Journal on Scientific Computing* 2000; **21**(5):1742–1749. Iterative methods for solving systems of algebraic equations (Copper Mountain, CO, 1998).
26. Parks ML, de Sturler E, Mackey G, Johnson DD, Maiti S. Recycling Krylov subspaces for sequences of linear systems. *SIAM Journal on Scientific Computing* 2006; **28**(5):1651–1674 (electronic).
27. Wang S, de Sturler E, Paulino GH. Large-scale topology optimization using preconditioned Krylov subspace methods with recycling. *International Journal for Numerical Methods in Engineering* 2007; **69**(12):2441–2468.
28. de Sturler E, Le C, Wang S, Paulino G. Large scale topology optimization using preconditioned Krylov subspace recycling and continuous approximation of material distribution. In *Multiscale and Functionally Graded Materials 2006* (*M&FGM 2006*), Paulino GH, Pindera M-J, Dodds Jr RH, Rochinha FA, Dave E, Chen L (eds), Oahu Island, Hawaii, 15–18 October 2006. *AIP Conference Proceedings*, 2008; 279–284.
29. Meurant G. On the incomplete Cholesky decomposition of a class of perturbed matrices. *SIAM Journal on Scientific Computing* 2001; **23**(2):419–429 (electronic). Copper Mountain Conference, 2000.
30. Benzi M, Bertaccini D. Approximate inverse preconditioning for shifted linear systems. *BIT* 2003; **43**(2):231–244.
31. Bertaccini D. Efficient preconditioning for sequences of parametric complex symmetric linear systems. *Electronic Transactions on Numerical Mathematics* 2004; **18**:49–64.
32. Duintjer Tebbens J, Tůma M. Efficient preconditioning of sequences of nonsymmetric linear systems. *SIAM Journal on Scientific Computing* 2007; **29**(5):1918–1941.
33. Duintjer Tebbens J, Tůma M. *Improving Triangular Preconditioner Updates for Nonsymmetric Linear Systems*. Lecture Notes in Computer Science, vol. 4818. Springer: New York, 2008; 737–744.
34. Birken P, Duintjer Tebbens J, Meister A, Tůma M. Preconditioner updates applied to CFD model problems. *Applied Numerical Mathematics* 2008; **58**(11):1628–1641.
35. Curtis AR, Powell MJD, Reid JK. On the estimation of sparse Jacobian matrices. *Journal of the Institute of Mathematics and its Applications* 1974; **13**:117–119.
36. Coleman TF, Moré JJ. Estimation of sparse Jacobian matrices and graph coloring problems. *SIAM Journal on Numerical Analysis* 1983; **20**:187–209.
37. Gebremedhin AH, Manne F, Pothen A. What color is your Jacobian? Graph coloring for computing derivatives. *SIAM Review* 2005; **47**:629–705.
38. Griewank A, Toint PL. On the unconstrained optimization of partially separable functions. *Nonlinear Optimization, 1981* (*Cambridge, 1981*). NATO Conference Series II: Systems Science. Academic Press: London, 1982; 301–312.
39. Cullum J, Tůma M. Matrix-free preconditioning using partial matrix estimation. *BIT Numerical Mathematics* 2006; **46**:711–729.
40. Garey MR, Johnson DS. *Computers and Intractability*: *A Guide to the Theory of NP-Completeness*. Freeman & Co.: New York, 1979.
41. Siefert C, de Sturler E. Probing methods for saddle-point problems. *Electronic Transactions on Numerical Analysis* 2006; **22**:163–183 (electronic).
42. Cuthill EH, McKee J. Reducing the bandwidth of sparse symmetric matrices. *Proceedings of the 24th National Conference of the ACM*. ACM Press: New York, 1969; 157–172.
43. Liu JWH, Sherman AH. Comparative analysis of the Cuthill–McKee and the reverse Cuthill–McKee ordering algorithms for sparse matrices. *SIAM Journal on Numerical Analysis* 1976; **13**:198–213.
44. Benzi M, Szyld DB, van Duin A. Orderings for incomplete factorization preconditioning of nonsymmetric problems. *SIAM Journal on Scientific Computing* 1999; **20**(5):1652–1670.
45. Lukšan L, Vlček J. Computational experience with globally convergent descent methods for large sparse systems of nonlinear equations. *Optimization Methods in Software* 1998; **8**(3–4):201–223.

DOI: 10.1002/nla

46. Lukšan L, Tůma M, Vlček J, Ramešová N, Šiška M, Hartman J, Matonoha C. UFO 2008—interactive system for universal functional optimization. *Technical Report V-1040*, downloadable from http://www.cs.cas.cz/luksan/ufo.html, ICS AS CR 2008.
47. Saad Y. ILUT: a dual threshold incomplete *LU* factorization. *Numerical Linear Algebra with Applications* 1994; **1**(4):387–402.
48. Hartmann S, Duintjer Tebbens J, Quint K, Meister A. Iterative solvers within sequences of large linear systems in non-linear structural mechanics. *ZAMM* 2009; **89**:711–728.

DOI: 10.1002/nla

Available online at www.sciencedirect.com
ScienceDirect
Computational Statistics & Data Analysis 52 (2007) 423–437
COMPUTATIONAL STATISTICS & DATA ANALYSIS
www.elsevier.com/locate/csda

Improving implementation of linear discriminant analysis for the high dimension/small sample size problem

Jurjen Duintjer Tebbens[a,*], Pavel Schlesinger[b]

[a] *Institute of Computer Science, Czech Academy of Sciences, Pod Vodarenskou vezi 2, 182 07 Prague 8, Czech Republic*
[b] *Institute of Formal and Applied Linguistics, Charles University, Malostranske namesti 25, 118 00 Prague 1, Czech Republic*

Available online 15 February 2007

Abstract

Classification based on Fisher's linear discriminant analysis (FLDA) is challenging when the number of variables largely exceeds the number of given samples. The original FLDA needs to be carefully modified and with high dimensionality implementation issues like reduction of storage costs are of crucial importance. Methods are reviewed for the high dimension/small sample size problem and the one closest, in some sense, to the classical regular approach is chosen. The implementation of this method with regard to computational and storage costs and numerical stability is improved. This is achieved through combining a variety of known and new implementation strategies. Experiments demonstrate the superiority, with respect to both overall costs and classification rates, of the resulting algorithm compared with other methods.

Keywords: Linear discriminant analysis; Numerical aspects of FLDA; Small sample size problem; Dimension reduction; Sparsity

1. Introduction

Fisher's linear discriminant analysis (FLDA) takes as one of the basic and first methods a prominent place in supervised classification tasks. Even in the presence of more advanced and sophisticated classification techniques and today's necessity to handle high dimensional data, FLDA has not left the minds of researchers. In this paper we address FLDA for the case where the number of variables largely exceeds the number of objects. In the literature this case has several names; in the pattern recognition community one mostly calls it the "small sample size" problem (see, e.g. Chen et al., 2000; Howland et al., 2006), in more general statistical literature like Hastie and Tibshirani (2003) we rather find the expression "$p > n$" or even "$p \gg n$" problem. To emphasize that the problem lies in the combination of many variables and few samples, we will here use "high dimension/small sample size" problem. Many classification strategies like nearest neighbor or support vector machines can be used to solve high dimension/small sample size problems. In this paper we restrict ourselves to FLDA-based approaches with emphasis on computational aspects. Generally, choosing the proper classification method is a state-of-the-art problem of analysis practise which we consider out of the scope of this paper. Also, we are aware of the general theoretical questions on applicability common to all methods for the high dimension/small sample size case that cope with singularity of covariance matrices. For these issues we refer the interested reader in the first place to Friedman (1989) and also to Hoffbeck and Landgrebe (1996) and

* Corresponding author.
E-mail addresses: tebbens@cs.cas.cz (J. Duintjer Tebbens), schlesinger@ufal.mff.cuni.cz (P. Schlesinger).

0167-9473/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2007.02.001

Bensmail and Celeux (1996) for *regularized discriminant analysis* and related strategies to solve the classical tasks of linear and quadratic discriminant analysis in the high dimension/small sample size case. For FLDA in particular we refer to the paper by Krzanowski et al. (1995).

Assuming we have decided to use an FLDA-based approach (a good reason may be FLDA's relative simplicity), we will first compare various criteria used to adapt FLDA to the high dimension/small sample size problem. Then the main part of the paper addresses implementation of the chosen criterion with emphasis on computational and storage costs. With high dimensional data, these issues are of crucial importance for the efficiency of the whole process; improved implementation may change a seemingly uncomputable problem to a perfectly solvable one. In addition, numerical stability plays an important role in the high dimension/small sample size case. We will propose an algorithm that exploits all advantageous implementation strategies we know of and we add some new ones. Our experiments show that, when thus implemented, FLDA has the potential to solve classification tasks with very high dimensional data.

In the remainder of this section we briefly recall original FLDA. Section 2 compares some of the best-known modifications of FLDA for the high dimension/small sample size problem. It concludes with the choice of the one we consider closest to the original FLDA. In Section 3 we present a very detailed description of our improved implementation. Numerical examples comparing it with other implementations are given in Section 4.

1.1. Classical FLDA

Consider a classification task with g groups, $g \geqslant 2$, and assume that n training objects (x_i, y_i) with $x_i \in \mathbb{R}^p$ and $y_i \in \{1, \ldots, g\}$ are available. Using the mean vector $\bar{x} = (1/n)\sum_{i=1}^{n} x_i$ and denoting by N_j the index set of objects in group j, by n_j the size of group j and by $\bar{x}_j = (1/n_j)\sum_{i \in N_j} x_i$ the corresponding group's mean vector, the *between-* and *within-group covariance matrix* $\mathbf{B}$ and $\mathbf{W}$, respectively, are defined by

$$\mathbf{B} = \frac{1}{g-1} \sum_{j=1}^{g} n_j (\bar{x}_j - \bar{x})(\bar{x}_j - \bar{x})^{\mathrm{T}}, \tag{1}$$

$$\mathbf{W} = \frac{1}{n-g} \sum_{j=1}^{g} \sum_{i \in N_j} (x_i - \bar{x}_j)(x_i - \bar{x}_j)^{\mathrm{T}}. \tag{2}$$

The rank of $\mathbf{B}$ is at most $\min(g-1, p)$, the rank of $\mathbf{W}$ is at most $\min(n, p)$.

Let us assume for the moment that $p < n$. Then *Fisher's criterion* (see, e.g. Duda et al., 2000; Ripley, 1996, originally Fisher, 1936) is to find, subsequently, at most $g-1$ transformation vectors c that have maximal separation ratio by solving the maximization problem

$$\max_{c \in \mathbb{R}^p,\, c \neq 0} \frac{c^{\mathrm{T}} \mathbf{B} c}{c^{\mathrm{T}} \mathbf{W} c}. \tag{3}$$

It can be translated to finding the largest eigenpairs of the generalized eigenproblem

$$(\mathbf{B} - \lambda \mathbf{W})c = 0, \tag{4}$$

which, in turn, can be transformed to a standard eigenproblem, for example $(\mathbf{W}^{-1}\mathbf{B} - \lambda \mathbf{I})c = 0$. Then the FLDA-reduced space of dimension i, $i < g$, is spanned by the eigenvectors corresponding to the i largest eigenvalues. They are ordered decreasingly according to the eigenvalues and are orthogonal to each other (see, e.g. Guo et al., 2003). Many applications just aim at dimension reduction and stop after mapping onto the FLDA-reduced space. In the original classification process, the simplest and most frequent way to classify is by assigning to the group j of the transformed group mean vector $(c_1, \ldots, c_i)^{\mathrm{T}} \bar{x}_j$ which is closest in the L_2-norm.

2. Fisher's criterion for the high dimension/small sample size problem

2.1. The $p > n$ case

When $p > n$ the covariance matrix $\mathbf{W}$ from (2) is singular. This makes the classical FLDA process we describe above hard to perform. The main problem is solving the generalized eigenproblem (4). We cannot transform it to a standard

eigenproblem anymore. This paper addresses cases where even $p \gg n$. Then the problem will be challenging to solve also with respect to storage and computational costs. These numerical aspects will be considered in Section 3.

With singular covariance matrices, the generalized eigenproblem can be ill-posed itself. We recall some facts from linear algebra to explain this (see, e.g. Bai et al., 2000). Eigenvectors c of (4) satisfy $\mathbf{B}c = \lambda \mathbf{W}c$, for some value λ. If c lies in the null space of $\mathbf{B}$ but not of $\mathbf{W}$, then λ is a zero eigenvalue. On the other hand, if c lies in the null space of $\mathbf{W}$ but not of $\mathbf{B}$, then we say λ is an infinite eigenvalue. If c does not lie in the null space of $\mathbf{B}$ and neither in the null space of $\mathbf{W}$, then λ must be finite and nonzero. If c lies in the common null space of $\mathbf{B}$ and $\mathbf{W}$, any value λ is an eigenvalue! In fact, in this case corresponding eigenvectors are not even defined (Bai et al., 2000). The presence of a common null space will make solving the eigenproblem (4) very challenging. For example, it is well known that the QZ-algorithm (Moler and Stewart, 1973) may solve generalized eigenproblems with singular matrices but suffers from numerical instability precisely with a common null space. Unfortunately, the covariance matrices $\mathbf{W}$ and $\mathbf{B}$ must have a common null space as soon as $n + g - 1 < p$.

Apart from the difficulties of solving (4), with a singular covariance matrix $\mathbf{W}$ Fisher's criterion itself to some extent loses its meaning: Transformation vectors c in the null space of $\mathbf{W}$ would lead to division by zero in (3). Here we focus on interpretation and modification of Fisher's criterion (3) in the $p \gg n$ case. Papers that address these issues include Chen et al. (2000), Cheng et al. (1992), Hong and Yang (1991), Howland et al. (2006), Li et al. (1999), Krzanowski et al. (1995) and Yang et al. (2000). We will briefly review and compare some of the most popular methods described in these papers. This will motivate our choice of the one modified criterion whose implementation we address afterwards.

2.2. Perturbation methods

One type of methods tries to transform (4) to a standard eigenproblem by overcoming the singularity of $\mathbf{W}$. A way to achieve this is by perturbation of the singular values of $\mathbf{W}$. More precisely, let $\mathbf{W} = \mathbf{QSQ}^{\mathrm{T}}$ be the singular value decomposition (SVD) of $\mathbf{W}$ (because $\mathbf{W}$ is symmetric it coincides with a spectral decomposition). Then the matrix of singular values $\mathbf{S}$ is replaced by $\mathbf{S} + \mathbf{D}$ where $\mathbf{D}$ is a diagonal matrix of small norm such that $\mathbf{S} + \mathbf{D}$ is nonsingular. Several choices of $\mathbf{D}$ are described in Cheng et al. (1992), Hong and Yang (1991), and Krzanowski et al. (1995). When $\tilde{\mathbf{W}}$ is the nonsingular matrix obtained by this kind of perturbation, then these methods determine the FLDA-vectors c by transforming the eigenproblem $(\mathbf{B} - \lambda\tilde{\mathbf{W}})c = 0$ to a standard eigenproblem. Working with the perturbed matrix $\tilde{\mathbf{W}}$ implies solving the *modified* criterion

$$\max_{c \in \mathbb{R}^p,\, c \neq 0} \frac{c^{\mathrm{T}}\mathbf{B}c}{c^{\mathrm{T}}\tilde{\mathbf{W}}c}. \tag{5}$$

Apart from the fact that it is not clear whether this method manages to solve Fisher's original criterion (3), it has the disadvantage that an optimal choice of the perturbation matrix $\mathbf{D}$ has to be determined, for example by cross-validation. While computing the spectral decomposition $\mathbf{QSQ}^{\mathrm{T}}$ of $\mathbf{W}$, the method asks for solving a symmetric p-dimensional eigenproblem with computational costs of order $\mathcal{O}(p^3)$ and storage costs of order $\mathcal{O}(p^2)$.

2.3. Methods exploiting the Moore–Penrose pseudo-inverse

A different way to obtain a standard eigenproblem results from considering the truncated SVD of $\mathbf{W}$. This method is mentioned, among others, in Cheng et al. (1992), Hong and Yang (1991), and Krzanowski et al. (1995), and is implemented in the statistical software `R`-environment (R Development Core Team, 2005) by the `lda`-function (see also Ripley, 1996; Venables and Ripley, 2002). Let the SVD of $\mathbf{W}$ be

$$\mathbf{W} = \mathbf{Q}\,\mathrm{diag}(s_1, \ldots, s_p)\mathbf{Q}^{\mathrm{T}}, \tag{6}$$

and let $|s_i| \leqslant \varepsilon$ for $i > r$ and some small tolerance $\varepsilon > 0$. Then if $\mathbf{Q}_r$ consists of the first r columns of $\mathbf{Q}$ and $\Lambda_r = \mathrm{diag}(s_1, \ldots, s_r)$, the truncated SVD of $\mathbf{W}$ is $\tilde{\mathbf{W}} = \mathbf{Q}_r \Lambda_r \mathbf{Q}_r^{\mathrm{T}}$. Instead of solving (4), these methods try to transform

$$(\mathbf{B} - \lambda\mathbf{Q}_r\, \Lambda_r \mathbf{Q}_r^{\mathrm{T}})c = 0 \tag{7}$$

to a standard eigenproblem by multiplication with the Moore–Penrose pseudo-inverse $\mathbf{Q}_r \Lambda_r^{-1} \mathbf{Q}_r^{\mathrm{T}}$ of $\tilde{\mathbf{W}}$. They solve, for example, the symmetric eigenproblem

$$(\Lambda_r^{-1/2} \mathbf{Q}_r^{\mathrm{T}} \mathbf{B} \mathbf{Q}_r \Lambda_r^{-1/2} - \lambda \mathbf{I}) c^* = 0, \tag{8}$$

where the desired eigenvectors c are obtained from

$$c = \mathbf{Q}_r \Lambda_r^{-1/2} c^* \tag{9}$$

and $\Lambda_r^{-1/2} = \mathrm{diag}(1/\sqrt{s_1}, \ldots, 1/\sqrt{s_r})$. The eigenproblem (8) is in general *not* equivalent to (7) because $\mathbf{Q}_r \mathbf{Q}_r^{\mathrm{T}} \neq \mathbf{I}$. Instead, the eigenvectors c obtained from solving (8) and (9) satisfy the equality $(\Lambda_r^{-1/2} \mathbf{Q}_r^{\mathrm{T}} \mathbf{B} \mathbf{Q}_r \mathbf{Q}_r^{\mathrm{T}} - \lambda \Lambda_r^{1/2} \mathbf{Q}_r^{\mathrm{T}}) c = 0$, hence by multiplying with $\mathbf{Q}_r \Lambda_r^{1/2}$ we have

$$(\mathbf{Q}_r \mathbf{Q}_r^{\mathrm{T}} \mathbf{B} \mathbf{Q}_r \mathbf{Q}_r^{\mathrm{T}} - \lambda \mathbf{Q}_r \Lambda_r \mathbf{Q}_r^{\mathrm{T}}) c = 0,$$

and one maximizes

$$\frac{c^{\mathrm{T}} \mathbf{Q}_r \mathbf{Q}_r^{\mathrm{T}} \mathbf{B} \mathbf{Q}_r \mathbf{Q}_r^{\mathrm{T}} c}{c^{\mathrm{T}} \mathbf{Q}_r \Lambda_r \mathbf{Q}_r^{\mathrm{T}} c}.$$

Here again, we do not know to what extent the original problem (3) is maximized. All we can do is measure the quality of the vectors c as approximate eigenvectors for the original eigenproblem (4).

Proposition 2.1. *Let us assume that $s_i = 0$ for all $i > r$ in the SVD* (6) *of* $\mathbf{W}$ *and let the last $p - r$ columns of* $\mathbf{Q}$, *corresponding to zero singular values*, *be denoted by* $\mathbf{Q}_z$. *Then the eigenpairs* $\{\lambda, c\}$ *defined through* (8) *and* (9) *satisfy*, *in the Euclidean norm*,

$$\|\mathbf{B}c - \lambda \mathbf{W} c\| = \|\mathbf{Q}_z^{\mathrm{T}} \mathbf{B} c\|.$$

Proof. We have

$$\mathbf{B}c - \lambda \mathbf{W} c = \mathbf{B}c - \lambda \mathbf{Q}_r \Lambda_r \mathbf{Q}_r^{\mathrm{T}} c = \mathbf{B} \mathbf{Q}_r \Lambda_r^{-1/2} c^* - \lambda \mathbf{Q}_r \Lambda_r^{1/2} c^*.$$

As $(\mathbf{Q}_r, \mathbf{Q}_z)$ is an orthonormal matrix,

$$\begin{aligned}
\|\mathbf{B}c - \lambda \mathbf{W} c\| &= \left\| \begin{pmatrix} \mathbf{Q}_r^{\mathrm{T}} \\ \mathbf{Q}_z^{\mathrm{T}} \end{pmatrix} (\mathbf{B} \mathbf{Q}_r \Lambda_r^{-1/2} c^* - \lambda \mathbf{Q}_r \Lambda_r^{1/2} c^*) \right\| \\
&= \left\| \begin{pmatrix} \Lambda_r^{1/2} \Lambda_r^{-1/2} \mathbf{Q}_r^{\mathrm{T}} \\ \mathbf{Q}_z^{\mathrm{T}} \end{pmatrix} (\mathbf{B} \mathbf{Q}_r \Lambda_r^{-1/2} c^* - \lambda \mathbf{Q}_r \Lambda_r^{1/2} c^*) \right\| \\
&= \left\| \begin{pmatrix} \Lambda_r^{1/2} (\Lambda_r^{-1/2} \mathbf{Q}_r^{\mathrm{T}} \mathbf{B} \mathbf{Q}_r \Lambda_r^{-1/2} c^* - \lambda c^*) \\ \mathbf{Q}_z^{\mathrm{T}} \mathbf{B} \mathbf{Q}_r \Lambda_r^{-1/2} c^* \end{pmatrix} \right\| = \left\| \begin{pmatrix} 0 \\ \mathbf{Q}_z^{\mathrm{T}} \mathbf{B} c \end{pmatrix} \right\|. \qquad \square
\end{aligned}$$

A similar result can easily be proven for perturbation methods. The previous proposition shows that methods exploiting the pseudo-inverse of $\mathbf{W}$ offer no room for improving the computed FLDA-vectors, their quality is fully determined by $\|\mathbf{Q}_z^{\mathrm{T}} \mathbf{B} c\|$. But they have the advantage they do not need to determine optimal perturbation parameters (only the truncation parameter ε is needed). Note that in (8) we solve an eigenproblem of dimension r, which is in general significantly less than p. However, the whole method asks for an initial p-dimensional spectral decomposition of $\mathbf{W}$ with computational costs of order $\mathcal{O}(p^3)$ and storage costs of order $\mathcal{O}(p^2)$.

2.4. A method based on the GSVD

Both types of methods we have described so far suffer from potential deterioration of the original eigenproblem (4) and hence Fisher's original criterion (3). A method that does not modify eigenproblem (4) is the LDA/GSVD (generalized singular value decomposition) method from Howland and Park (2004), Howland et al. (2003,2006) and Kim et al. (2005). It extracts the eigenvectors of (4) needed for FLDA. This is achieved by using the GSVD

(Paige and Saunders, 1981; Golub and van Loan, 1996). Leaving aside the details, with a numerically stable algorithm for the GSVD we can find diagonal matrices with nonnegative entries $\mathbf{S}_\alpha = \mathrm{diag}(\alpha_1, \ldots, \alpha_t)$ and $\mathbf{S}_\beta = \mathrm{diag}(\beta_1, \ldots, \beta_t)$ and a nonsingular matrix $\mathbf{C} \in \mathbb{R}^{p\times p}$ such that

$$\mathbf{B} = \mathbf{C}^{-\mathrm{T}} \begin{pmatrix} \mathbf{S}_\alpha & 0 \\ 0 & 0 \end{pmatrix} \mathbf{C}^{-1}, \quad \mathbf{W} = \mathbf{C}^{-\mathrm{T}} \begin{pmatrix} \mathbf{S}_\beta & 0 \\ 0 & 0 \end{pmatrix} \mathbf{C}^{-1},$$

with $\mathbf{S}_\alpha + \mathbf{S}_\beta = \mathbf{I}_t$ and $t \leqslant n + g$. This implies the first t columns c_i of $\mathbf{C}$ are eigenvectors for (4) and satisfy

$$\beta_i \mathbf{B} c_i = \alpha_i \mathbf{W} c_i.$$

If β_i is zero, the eigenvectors lie in the null space of $\mathbf{W}$ but not in the null space of $\mathbf{B}$. Then the within-group variance $c_i^{\mathrm{T}} \mathbf{W} c_i$ is zero and hence minimal. For this reason, the LDA/GSVD method chooses these vectors as the leading FLDA-transformation vectors. The remaining ones are chosen according to the ratio α_i/β_i; those for which

$$\frac{\alpha_i}{\beta_i} = \frac{c_i^{\mathrm{T}} \mathbf{B} c_i}{c_i^{\mathrm{T}} \mathbf{W} c_i}$$

is largest are chosen first. This corresponds to Fisher's original criterion (3). The last $p - t$ columns of $\mathbf{C}$ span the common null space of $\mathbf{B}$ and $\mathbf{W}$. In the common null space both between-group and within-group variance are zero. Therefore, no vectors from this space are used. The LDA/GSVD method can be implemented attractively by exploiting the special structure of the covariance matrices (we explain this in Section 3). This causes computational costs to be reduced to $\mathcal{O}(pn^2) + \mathcal{O}(n^3)$ and storage costs to $\mathcal{O}(p(n+g))$. In addition, the method offers a mathematical framework that helps in better understanding the high dimension/small sample size problem, see, e.g. Howland et al. (2003).

2.5. The null space method

We see that in the LDA/GSVD method the criterion (3) is modified by separating vectors for which $c_i^{\mathrm{T}} \mathbf{W} c_i$ is zero from those that yield a finite ratio $(c_i^{\mathrm{T}} \mathbf{B} c_i)/(c_i^{\mathrm{T}} \mathbf{W} c_i)$. The classical null space method (see, e.g. Chen et al., 2000 or the so-called zero-variance discrimination method in Krzanowski et al., 1995) fully concentrates on the null space of $\mathbf{W}$. As in LDA/GSVD, this is motivated by the fact that in this space within-group variance is minimal. The null space method simply modifies (3) as

$$\max_{c\in\mathbb{R}^p,\, \mathbf{W}c=0} c^{\mathrm{T}} \mathbf{B} c. \tag{10}$$

Of course, we maximize over vectors c with unit norm. The criterion leads to a standard eigenproblem in the null space of $\mathbf{W}$. It should be noticed that this null space has large dimension if $p \gg n$. As the rank of $\mathbf{W}$ is at most n, the null space has dimension at least $p - n$, which is just a little less than p. Therefore, this method finds, in addition to the spectral decomposition of $\mathbf{W}$, another large spectral decomposition and may be very time-consuming. Dominating computational costs are of order $\mathcal{O}(p^3)$; storage costs are of order $\mathcal{O}(p^2)$.

2.6. An intuitively reasonable criterion

Of all methods we discussed, Fisher's original idea to minimize within-group variance and maximize between-group variance seems best realized by the criterion (10). However, the null space method may choose vectors from the common null space where $c^{\mathrm{T}} \mathbf{B} c$ is zero as well: We look for $g - 1$ transformation vectors in total and the number of vectors in the null space of $\mathbf{W}$ that are not in the common null space can be less than $g - 1$. We avoid this by using a combined criterion that can be described as follows.

Transformation vectors from the null space of $\mathbf{W}$ give the "maximal" ratio $c^{\mathrm{T}} \mathbf{B} c / c^{\mathrm{T}} \mathbf{W} c = \infty$. As in the previous two methods, we choose them as leading transformation vectors because their within-group variance $c^{\mathrm{T}} \mathbf{W} c = 0$ is minimal. We order them according to their between-group variance, i.e. we use the criterion from the null space method,

$$\max_{c\in\mathbb{R}^p,\, \mathbf{W}c=0} c^{\mathrm{T}} \mathbf{B} c. \tag{11}$$

However, transformation vectors for which the maximum in (11) is zero are not interesting anymore; their between-group variance is minimal, hence they do not contribute to discrimination. Therefore, we select with criterion (11) only transformation vectors with nonzero between-group variance. If this does not yield enough (mutually orthogonal) transformation vectors, we leave the null space of $\mathbf{W}$ and select the next transformation vectors in the complement of the null space of $\mathbf{W}$. Here of course, the ratio $c^{\mathrm{T}}\mathbf{B}c/c^{\mathrm{T}}\mathbf{W}c$ is always finite and we can use the original criterion

$$\max_{c\in\mathbb{R}^p,\ \mathbf{W}c\neq 0} \frac{c^{\mathrm{T}}\mathbf{B}c}{c^{\mathrm{T}}\mathbf{W}c}. \tag{12}$$

The intuitively reasonable, combined criterion (11–12), which follows logically from the previously considered criteria, has been proposed, for example, in Yang and Yang (2003). We believe it reproduces Fisher's original idea best and we will use it in our implementation too. Our experiments seem to indicate that it leads to at least as powerful discrimination as other criteria.

3. Efficient implementation

The main focus of this paper is efficient implementation of (11) and (12). We have tried to combine as many clever strategies that are known as possible in order to minimize the overall costs and reduce numerical instability. In addition, we introduce some new ideas that make the algorithm even faster. We begin with two commonly used tools: Writing $\mathbf{B}$ and $\mathbf{W}$ as products of rectangular matrices and elimination of the common null space.

3.1. Exploiting the special structure of covariance matrices

The within-group and between-group covariance matrices $\mathbf{W}$ and $\mathbf{B}$ are both full matrices of dimension p. In many applications p is just too large to be able to store $2p^2$ matrix entries. For example, when $p = 10\,000$, which is realistic among others in modern document classification tasks, then $\mathbf{B}$ and $\mathbf{W}$ take already 1.6 GB to be stored in double precision arithmetic. Also, computations with these large matrices are rather expensive. In order to work efficiently with covariance matrices one commonly takes advantage of the fact that they can be written as a product of one and the same rectangular matrix. Let $\mathbf{X} \in \mathbb{R}^{n\times p}$ be the sample matrix whose ith row contains the ith training object and let $\mathbf{1_n} = (1, 1, \ldots, 1)^{\mathrm{T}} \in \mathbb{R}^n$. Furthermore, let $\mathbf{M} \in \mathbb{R}^{g\times p}$ be the group mean matrix whose jth row contains $\bar{x}_j^{\mathrm{T}}$ and let $\mathbf{G} \in \mathbb{R}^{n\times g}$ be the group coding matrix. If the ith object belongs to group j, $\mathbf{G}_{i,j} = 1$ and $\mathbf{G}_{i,k} = 0$ for $k \neq j$. Then (see, e.g. Venables and Ripley, 2002),

$$\mathbf{W} = \frac{1}{n-g}\sum_{j=1}^{g}\sum_{i\in N_j}(x_i - \bar{x}_j)(x_i - \bar{x}_j)^{\mathrm{T}} = \frac{(\mathbf{X} - \mathbf{GM})^{\mathrm{T}}(\mathbf{X} - \mathbf{GM})}{n-g}. \tag{13}$$

The matrix $\mathbf{B}$ can be written as

$$\mathbf{B} = \frac{1}{g-1}\sum_{j=1}^{g} n_j(\bar{x}_j - \bar{x})(\bar{x}_j - \bar{x})^{\mathrm{T}} = \frac{(\mathbf{GM} - \mathbf{1_n}\bar{x}^{\mathrm{T}})^{\mathrm{T}}(\mathbf{GM} - \mathbf{1_n}\bar{x}^{\mathrm{T}})}{g-1} \tag{14}$$

and also as (see, e.g. Kim et al., 2005)

$$\mathbf{B} = \frac{(\tilde{\mathbf{D}}(\mathbf{M} - \mathbf{1}_g\bar{x}^{\mathrm{T}}))^{\mathrm{T}}\tilde{\mathbf{D}}(\mathbf{M} - \mathbf{1}_g\bar{x}^{\mathrm{T}})}{g-1}, \tag{15}$$

where $\tilde{\mathbf{D}} = \mathrm{diag}(1/n_1, \ldots, 1/n_g)$.

In all cases a covariance matrix is decomposed into two rectangular matrices that are each others transposed. The number of rows of the right rectangles n (or even g for (15)) is by assumption much smaller than the number of columns p. It is advantageous to store only one rectangular part and replace computations with the covariance matrices by computations with their rectangles. A very simple example is the product $z = \mathbf{W}v$ of $\mathbf{W}$ with a vector v. It can be

computed by first forming the n-dimensional vector $y = (\mathbf{X} - \mathbf{GM})v$ and then putting

$$z = \frac{(\mathbf{X} - \mathbf{GM})^{\mathrm{T}} y}{n - g}.$$

As the direct product $z = \mathbf{W}v$ costs $2p^2$ floating point operations and the small products cost $2pn$ operations each, computational costs are reduced as soon as $n < p/2$. We consider efficient multiplications more in detail in Section 3.4. If we manage to restrict all computations needed with $\mathbf{B}$ and $\mathbf{W}$ to their rectangular factors in similar ways, we avoid storing $\mathbf{B}$ and $\mathbf{W}$. This has been successfully accomplished in the LDA/GSVD method and in the lda()-function implemented in the R-environment (R Development Core Team, 2005). Our implementation will also take advantage of the special structure of the covariance matrices.

3.2. Elimination of the common null space

A technique that has many advantages in FLDA-based computations is elimination of the common null space of $\mathbf{B}$ and $\mathbf{W}$. It is justified by the fact that vectors c in the common null space do not contribute to discrimination because $c^{\mathrm{T}}\mathbf{B}c = 0 = c^{\mathrm{T}}\mathbf{W}c$ (see, e.g. Yang and Yang, 2003). The common null space can be eliminated very efficiently by considering the *total* covariance matrix. This matrix is defined as

$$\mathbf{T} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^{\mathrm{T}} = \frac{1}{n-1}(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})(\mathbf{X} - \mathbf{1}_{\mathbf{n}}\bar{x}^{\mathrm{T}}). \tag{16}$$

The following relation between the covariance matrices holds:

$$(n-1)\mathbf{T} = (g-1)\mathbf{B} + (n-g)\mathbf{W},$$

see, e.g. Howland et al. (2003). If we drop the denominators in (13), (14) and (16), the relation translates to

$$\mathbf{T} = \mathbf{B} + \mathbf{W}. \tag{17}$$

Discarding the denominators has no influence on the computations we perform. Hence from now on we consider unscaled covariance matrices for simplicity. As a consequence of (17) we have the following well-known lemma (Yang and Yang, 2003). For completeness we also give its proof.

Lemma 1. *The common null space of* $\mathbf{B}$ *and* $\mathbf{W}$ *is the null space of* $\mathbf{T}$.

Proof. A vector $v \in \mathbb{R}^p$ lies in the null space of $\mathbf{T}$ if and only if $v^{\mathrm{T}}\mathbf{T}v = 0$. This is readily seen from

$$v^{\mathrm{T}}\mathbf{T}v = 0 \;\Rightarrow\; v^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})(\mathbf{X} - \mathbf{1}_{\mathbf{n}}\bar{x}^{\mathrm{T}})v = 0 \;\Rightarrow\; \|(\mathbf{X} - \mathbf{1}_{\mathbf{n}}\bar{x}^{\mathrm{T}})v\|^2 = 0,$$

the other direction is trivial. The same holds for $\mathbf{W}$ and $\mathbf{B}$ because they can be written as (13) and (14), respectively (here without the denominators). With (17) and the fact that $\mathbf{W}$ and $\mathbf{B}$ are positive semi-definite we have

$$v^{\mathrm{T}}\mathbf{T}v = 0 \;\Leftrightarrow\; v^{\mathrm{T}}(\mathbf{B} + \mathbf{W})\,v = 0 \;\Leftrightarrow\; v^{\mathrm{T}}\mathbf{B}v = 0 \quad \text{and} \quad v^{\mathrm{T}}\mathbf{W}v = 0. \qquad \square$$

In other words, the complement of the common null space of $\mathbf{B}$ and $\mathbf{W}$ is spanned by the eigenvectors of $\mathbf{T}$ which correspond to nonzero eigenvalues of $\mathbf{T}$. We show below that these eigenvectors can be computed inexpensively. Note that restriction to the eigenvectors of nonzero eigenvalues of $\mathbf{T}$ is nothing but performing a classical PCA as a preprocessing step and including all principal components explaining the full 100% of total variability (Yang and Yang, 2003).

If the total covariance matrix $\mathbf{T}$ has rank q where $q \leqslant n$, this preprocessing reduces the original p-dimensional problem to the dimension q. As we assume $p \gg n$, the benefit is considerable. Another important advantage of elimination of the common null space is that it enhances numerical stability of algorithms for generalized eigenproblems, see for example Parlett (1998).

The eigenvectors of $\mathbf{T}$ corresponding to nonzero eigenvalues can be computed very efficiently with the following lemma.

Lemma 2. *Let $\mathbf{Z} \in \mathbb{R}^{n\times p}$ with $n < p$, let the diagonal matrix $\mathbf{D}_1$ contain the nonzero eigenvalues of $\mathbf{Z}\mathbf{Z}^{\mathrm{T}} \in \mathbb{R}^{n\times n}$ and let the columns of $\mathbf{V}_1$ contain the corresponding eigenvectors. Then the normalized eigenvectors for nonzero eigenvalues of $\mathbf{Z}^{\mathrm{T}}\mathbf{Z} \in \mathbb{R}^{p\times p}$ are given by the columns of $\mathbf{Z}^{\mathrm{T}}\mathbf{V}_1\mathbf{D}_1^{-1/2}$.*

Proof. See Johnson and Wichern (1998). □

This lemma is widely used in PCA computations. It says we can extract eigenvectors of the p-dimensional matrix $\mathbf{T} = (\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})(\mathbf{X} - \mathbf{1}_{\mathbf{n}}\bar{x}^{\mathrm{T}})$ by forming the n-dimensional spectral decomposition

$$(\mathbf{X} - \mathbf{1}_{\mathbf{n}}\bar{x}^{\mathrm{T}})(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}}) = \mathbf{V}\mathbf{D}\mathbf{V}^{\mathrm{T}}, \tag{18}$$

where $\mathbf{D}$ is a diagonal matrix containing the eigenvalues in decreasing order. Let it have the form

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_1 & 0 \\ 0 & 0 \end{pmatrix},$$

where $\mathbf{D}_1 \in \mathbb{R}^{q\times q}$ is nonsingular. If we collect in $\mathbf{V}_1$ the eigenvectors of $\mathbf{V}$ corresponding to the nonzero eigenvalues then the complement of the null space of $\mathbf{T}$ is spanned by the q orthonormal columns of

$$(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})\mathbf{V}_1\mathbf{D}_1^{-1/2}. \tag{19}$$

Note that the only computation depending on p needed to find the complement of the common null space (19) is multiplication with $(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})$. In Section 3.4 we show how to circumvent the high costs of this multiplication.

3.3. Efficient computations in the complement of the common null space

We denote the projections of the matrices $\mathbf{B}$, $\mathbf{W}$ and $\mathbf{T}$ onto the complement of the common null space, which is spanned by the columns of (19), by $\overline{\mathbf{B}}$, $\overline{\mathbf{W}}$ and $\overline{\mathbf{T}}$, respectively. We now show how we solve (4) in the complement of the common null space. To facilitate computations we use the following simple lemma from linear algebra (Bai et al., 2000).

Lemma 3. *Any generalized eigenvector c satisfying $\mathbf{Y}c = \mu(\mathbf{Y} + \mathbf{Z})c$ for some eigenvalue $\mu \in \mathbb{R}$ satisfies $\mathbf{Y}c = (\mu/(1-\mu))\mathbf{Z}c$, where the corresponding eigenvalue is infinite if $\mu = 1$.*

Hence with $\overline{\mathbf{T}} = \overline{\mathbf{B}} + \overline{\mathbf{W}}$, any eigenvector c with

$$\overline{\mathbf{B}}c = \mu\overline{\mathbf{T}}c \tag{20}$$

satisfies

$$\overline{\mathbf{B}}c = \lambda\overline{\mathbf{W}}c, \tag{21}$$

where $\lambda = \mu/(1-\mu)$. This means that the eigenvectors of (20) are the same as those of (21). As we need in FLDA only the eigenvectors, we can solve (20) instead of (21), provided we select the eigenvectors correctly. Infinite eigenvalues of (21) take the value 1 in (20) and finite eigenvalues change to eigenvalues that are smaller than 1.

Using (20) instead of (21) has been proposed among others in Cheng et al. (1992), and Hong and Yang (1991) in order to modify Fisher's criterion. We are here interested in two important implementational advantages which to our knowledge the literature is not fully aware of. The first one is that $\overline{\mathbf{T}}$ is nonsingular because it is the restriction of $\mathbf{T}$ to the complement of its own null space. Hence (20) can be transformed to a standard eigenproblem. The second advantage is that (20) takes a particularly simple form.

Lemma 4. *The projection $\overline{\mathbf{T}}$ of $\mathbf{T}$ to the complement of the common null space is the nonsingular diagonal matrix $\mathbf{D}_1$.*

Proof. Using (19), we have

$$\begin{aligned}\overline{\mathbf{T}} &= ((\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})\mathbf{V}_1\mathbf{D}_1^{-1/2})^{\mathrm{T}}\mathbf{T}((\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})\mathbf{V}_1\mathbf{D}_1^{-1/2}) \\ &= \mathbf{D}_1^{-1/2}\mathbf{V}_1^{\mathrm{T}}(\mathbf{X} - \mathbf{1}_{\mathbf{n}}\bar{x}^{\mathrm{T}})(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})(\mathbf{X} - \mathbf{1}_{\mathbf{n}}\bar{x}^{\mathrm{T}})(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})\mathbf{V}_1\mathbf{D}_1^{-1/2}.\end{aligned}$$

From (18) we obtain $(\mathbf{X} - \mathbf{1}_{\mathbf{n}}\bar{x}^{\mathrm{T}})(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})\mathbf{V}_1 = \mathbf{V}_1\mathbf{D}_1$ and $\overline{\mathbf{T}}$ simplifies to the diagonal matrix $\mathbf{D}_1$. □

Hence we do not need to compute $\overline{\mathbf{T}}$ at all. For $\overline{\mathbf{B}}$, we have

$$\overline{\mathbf{B}} = ((\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})\mathbf{V}_1\mathbf{D}_1^{-1/2})^{\mathrm{T}}\mathbf{B}((\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})\mathbf{V}_1\mathbf{D}_1^{-1/2}).$$

As in (14), we can write $\overline{\mathbf{B}}$ as $\overline{\mathbf{B}} = \mathbf{B}_1^{\mathrm{T}}\mathbf{B}_1$ where

$$\mathbf{B}_1 = (\mathbf{GM} - \mathbf{1}_{\mathbf{n}}\bar{x}^{\mathrm{T}})(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})\mathbf{V}_1\mathbf{D}_1^{-1/2}.$$

Thus (20) takes the form

$$(\mathbf{B}_1^{\mathrm{T}}\mathbf{B}_1 - \mu\mathbf{D}_1)c = 0. \tag{22}$$

In our implementation we will transform (22) to the symmetric standard eigenproblem

$$(\mathbf{D}_1^{-1/2}\mathbf{B}_1^{\mathrm{T}}\mathbf{B}_1\mathbf{D}_1^{-1/2})c^* = \mu c^*, \quad c = \mathbf{D}_1^{-1/2}c^*. \tag{23}$$

We emphasize that transformation to this standard eigenproblem is possible because we base our computations on (20) instead of (21). Note that we are not interested in all q eigenpairs but only in the $g - 1$ leading ones.

The next and last step is solving the maximization problems (11) and (12) from Section 2.6 in the complement of the common null space. In contrast with other implementations (see, e.g. Yang and Yang, 2003) we do not solve these maximization problems separately but we extract all the needed vectors from (22). This makes the implementation faster and simpler. As explained by Lemma 3, eigenvalues $\mu = 1$ for (20) are infinite eigenvalues λ for (21). Hence the corresponding eigenvectors lie in the null space of $\overline{\mathbf{W}}$ and we will use them to solve the first part (11) of our criterion

$$\max_{c\in\mathbb{R}^p,\, \overline{\mathbf{W}}c=0} c^{\mathrm{T}}\overline{\mathbf{B}}c. \tag{24}$$

The computed eigenvectors for $\mu = 1$ necessarily form a basis for this null space because of the following lemma.

Lemma 5. *The null space of* $\overline{\mathbf{W}}$ *has dimension at most* $g - 1$.

Proof. Assume the dimension of the null space of $\overline{\mathbf{W}}$ is larger than $g - 1$. Then there exists at least one vector v in this null space with $v^{\mathrm{T}}\overline{\mathbf{B}}v = 0$ because the rank of $\overline{\mathbf{B}}$ is at most $g - 1$. Hence v lies in the common null space, which is a contradiction to the definition of the null space of $\overline{\mathbf{W}}$. □

To solve the maximization problem (24) correctly we need an *orthogonal* basis of the null space of $\overline{\mathbf{W}}$. Let us collect computed eigenvectors for $\mu = 1$ in a matrix $\mathbf{V}_2$. Then we propose to compute the reduced QR-decomposition

$$\mathbf{V}_2 = \mathbf{QR},$$

i.e. $\mathbf{Q}$ is orthogonal and rectangular, $\mathbf{R}$ is upper triangular and square with dimension equal to the number of columns of $\mathbf{V}_2$. This QR-decomposition is very cheap because $\mathbf{V}_2$ has few columns (namely, less than g). The columns of $\mathbf{Q}$ form an orthogonal basis of the null space of $\overline{\mathbf{W}}$ and we compute the eigenvectors $\tilde{c}$ of $\mathbf{Q}^{\mathrm{T}}\mathbf{B}_1^{\mathrm{T}}\mathbf{B}_1\mathbf{Q}$. Then the (ordered) vectors $\mathbf{Q}\tilde{c}$ maximize $c^{\mathrm{T}}\overline{\mathbf{B}}c$ subject to $\overline{\mathbf{W}}c = 0$.

For the second part (12) of our criterion, we consider the remaining eigenvalues from (22). They satisfy $\mu < 1$ and are finite eigenvalues λ for (21). The corresponding eigenvectors lie in the complement of the null space of $\overline{\mathbf{W}}$ and maximize Fisher's original criterion

$$\frac{c^{\mathrm{T}}\overline{\mathbf{B}}c}{c^{\mathrm{T}}\overline{\mathbf{W}}c}. \tag{25}$$

in the complement of the common null space.

If we store first the vectors obtained from (24) and then those from (25) in a matrix $\mathbf{C}$, we return to the original p-dimensional space by multiplying $\mathbf{C}$ with $(\mathbf{X} - \mathbf{1_n}\bar{x}^T)^T\mathbf{V}_1\mathbf{D}_1^{-1/2}$ from (19) and thus obtain the final FLDA-transformation vectors. The overall algorithm has the following form.

Algorithm 1. A fast algorithm to solve the FLDA-based criterion (11)–(12).

(1) Compute the spectral decomposition of $(\mathbf{X} - \mathbf{1_n}\bar{x}^T)(\mathbf{X}^T - \bar{x}\mathbf{1_n}^T)$; store the nonzero eigenvalues in the diagonal matrix $\mathbf{D}_1$ and the corresponding eigenvectors in $\mathbf{V}_1$.
(2) Compute $\mathbf{B}_1 = ((\mathbf{GM} - \mathbf{1_n}\bar{x}^T)(\mathbf{X}^T - \bar{x}\mathbf{1_n}^T))\mathbf{V}_1\mathbf{D}_1^{-1}$.
(3) Compute the eigenvectors of the $g-1$ largest eigenvalues of $\mathbf{B}_1^T\mathbf{B}_1$ and multiply them with $\mathbf{D}_1^{-1/2}$.
(4) If any, collect the eigenvectors for the eigenvalue 1 in $\mathbf{V}_2$ and
  (a) compute the reduced QR-decomposition $\mathbf{V}_2 = \mathbf{QR}$;
  (b) compute the eigenvectors of $\mathbf{Q}^T\mathbf{D}_1^{1/2}\mathbf{B}_1^T\mathbf{B}_1\mathbf{D}_1^{1/2}\mathbf{Q}$;
  (c) multiply them with $\mathbf{Q}$ and substitute the eigenvectors for the eigenvalue 1 from step (3) with these vectors.
(5) Multiply the vectors obtained from step (3) and possibly step (4) with $(\mathbf{X}^T - \bar{x}\mathbf{1_n}^T)\mathbf{V}_1\mathbf{D}_1^{-1/2}$ and normalize them.

3.4. Remarks on the algorithm

Here we address two more issues that may accelerate the overall algorithm: Clever multiplication of matrices and the usage of so-called sparse methods to solve the eigenproblems. We consider here the possibilities offered by MATLAB (MathWorks, Inc., 1984–2005). Sufficient experience with LAPACK (Anderson et al., 2000) and similar packages to implement our algorithm seems unrealistic for the average statistician. On the other hand, the relatively simple programming language of MATLAB has an advantage over the R-environment (R Development Core Team, 2005) that is important in the context of FLDA-based classification: Working with sparse matrices is very well-integrated. Storage of matrices in sparse format is possible without loading special packages and, more important, MATLAB contains so-called sparse algorithms for eigenvalue computations with sparse matrices. Such algorithms are not yet available in the R-environment.

MATLAB basically offers two functions to solve eigenproblems numerically: The command eig uses a so-called direct method and eigs uses a sparse method. At the end of the computation, a direct method has found all eigenpairs, but no eigenpairs are available during the process. Computational costs are of order m^3 if m is the dimension of the eigenproblem. Direct methods are backward stable, i.e. the computed pairs are exact eigenpairs for a different yet close eigenproblem. Accuracy of an eigenvector is endangered only when the corresponding eigenvalue lies close to other eigenvalues. Sparse methods, on the other hand, are advantageous if multiplication of vectors with the involved matrix is inexpensive and if we need only a few eigenvalues and eigenvectors. They compute one eigenvalue at a time and can be stopped after a predefined number of eigenpairs has been found. Computational costs depend on the sparsity of the matrix and the number of eigenvalues that is needed. They are not backward stable and convergence of computed eigenpairs to the wanted eigenpairs is not guaranteed.

In our algorithm we solve eigenproblems in steps (1), (3) and (4b). The first and third one need all eigenpairs, hence we recommend to use the eig command to solve them. In step (3) we need the leading $g-1$ eigenpairs of a q-dimensional problem. If the dimension q of the common null space is clearly larger than $g-1$, using a sparse method with eigs is in general much faster than using eig. In our implementation we used a sparse method in step (3).

To further reduce computational and storage costs we propose to perform the multiplications of $p \times n$ matrices in Algorithm 1 as follows. In step (1) we recommend to directly form the sum $\mathbf{X}\mathbf{X}^T - \mathbf{X}\bar{x}\mathbf{1_n}^T - \mathbf{1_n}\bar{x}^T\mathbf{X}^T + \|\bar{x}\|^2\mathbf{1_n}\mathbf{1_n}^T$. If we would form the factor $(\mathbf{X} - \mathbf{1_n}\bar{x}^T)$ we would create an additional $p \times n$ matrix to be stored; in our case we store only matrices of dimension n. Because $\bar{x} = \mathbf{X}^T\mathbf{1_n}/n$ we need to compute only $\tilde{\mathbf{X}} \equiv \mathbf{X}\mathbf{X}^T$ and the vector $\tilde{\mathbf{X}}_1 \equiv \tilde{\mathbf{X}}\mathbf{1_n}$; then the wanted sum is

$$\tilde{\mathbf{X}} - (\tilde{\mathbf{X}}_1\mathbf{1_n}^T)/n - (\mathbf{1_n}\tilde{\mathbf{X}}_1^T)/n + \|\bar{x}\|^2\mathbf{1}_n\mathbf{1}_n^T. \tag{26}$$

It is easy to see that $\|\bar{x}\|^2$ can be computed as $\mathbf{1_n}^T\tilde{\mathbf{X}}\mathbf{1_n}$ divided by n^2. Computational costs are dominated by the computation of $\tilde{\mathbf{X}}$. Although in general this computation involves $2n^2p$ operations (Golub and van Loan, 1996), here it may be significantly reduced because in many applications (protein fold prediction, text document classification, etc.)

the sample matrix $\mathbf{X}$ has many zero entries. If we store it as a sparse matrix and the number of nonzero entries in row i is denoted by nnz_i, then forming the matrix vector product $\mathbf{X}v$ for some vector $v \in \mathbb{R}^p$ costs $2\sum_{i=1}^{n} \mathsf{nnz}_i$ operations in MATLAB. With $\mathsf{nnz} = \sum_{i=1}^{n} \mathsf{nnz}_i$ the total costs of computing $\tilde{\mathbf{X}}$ are $2\mathsf{nnz}n$ at most and can be significantly less than $2n^2p$.

Similarly, it is important to compute $\mathbf{B}_1$ efficiently in step (2). Again, if we would form $(\mathbf{X} - \mathbf{1_n}\bar{x}^{\mathrm{T}})^{\mathrm{T}}\mathbf{V}_1\mathbf{D}_1^{-1/2}$ we would create an additional $p \times n$ matrix to be stored. This is avoided by computing first the product

$$(\mathbf{GM} - \mathbf{1_n}\bar{x}^{\mathrm{T}})(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1_n}^{\mathrm{T}}) = \mathbf{GMX}^{\mathrm{T}} - \mathbf{GM}\bar{x}\mathbf{1_n}^{\mathrm{T}} - (\mathbf{1_n}\tilde{\mathbf{X}}_1^{\mathrm{T}})/n + \|\bar{x}\|^2\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}},$$

which is a sum of $n \times n$ matrices. Note that the last two terms have been computed already in step (1) in (26). Moreover, the group coding matrix $\mathbf{G}$ is always sparse and we can write $\mathbf{M}$ as $\mathbf{M} = \tilde{\mathbf{D}}\mathbf{G}^{\mathrm{T}}\mathbf{X}$ with the diagonal matrix $\tilde{\mathbf{D}}$ from (15). With $\mathbf{M}_1 \equiv \tilde{\mathbf{D}}\mathbf{G}^{\mathrm{T}}\tilde{\mathbf{X}}$ we have

$$(\mathbf{GM} - \mathbf{1_n}\bar{x}^{\mathrm{T}})(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1_n}^{\mathrm{T}}) = \mathbf{G}(\mathbf{M}_1(\mathbf{I} - \mathbf{1_n}\mathbf{1_n}^{\mathrm{T}}/n)) - (\mathbf{1_n}\tilde{\mathbf{X}}_1^{\mathrm{T}})/n + \|\bar{x}\|^2\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}}.$$

Finally, we need in step (5) the product $(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1_n}^{\mathrm{T}})\mathbf{V}_1\mathbf{D}_1^{-1/2}$. For the same reasons as in step (2), it is best computed as

$$\mathbf{X}^{\mathrm{T}}\left(\mathbf{V}_1\mathbf{D}_1^{-1/2} - \mathbf{1}_n\left(\frac{\mathbf{1}_n^{\mathrm{T}}\mathbf{V}_1\mathbf{D}_1^{-1/2}}{n}\right)\right).$$

All together, we see that we do not even need to store the rectangular factors of the covariance matrices. It suffices to store $\mathbf{X}$, $\tilde{\mathbf{X}}$, $\tilde{\mathbf{X}}_1$, $\mathbf{G}$ and $\mathbf{M}_1$.

3.5. Concluding remarks

We conclude this section with a brief summary of the influence on overall costs of the techniques discussed here. In our algorithm, computational costs are dominated by the products with $p \times n$ matrices and the eigenproblem in step (1). In general the products ask for $\mathcal{O}(pn^2)$ operations. In many cases, however, the data matrix is sparse and the costs will be of order $\mathcal{O}(\mathsf{nnz}n)$, where nnz is the number of nonzero entries in $\mathbf{X}$. The eigenproblem in step (1) needs $\mathcal{O}(n^3)$ operations. We have to store only one $p \times n$ matrix, namely $\mathbf{X}$. The final FLDA-transformation vectors can be stored in the first columns of $\mathbf{X}$. Hence memory requirements are of order $\mathcal{O}(pn)$.

The methods from Section 2 can partially profit from our implementation strategies too. This would give the following rough costs for the individual methods. Perturbation methods can take advantage of the structure of covariance matrices and of a sparse method to find the $g-1$ largest eigenvalues. For dense sample matrices this gives a complexity of order $\mathcal{O}(p^2n)$; storage costs are of order $\mathcal{O}(p^2)$. Methods exploiting the Moore–Penrose pseudo-inverse can be implemented with elimination of the common null space, making usage of the special structure of covariance matrices and a sparse method for the eigenproblem in the complement of the null space of $\overline{\mathbf{W}}$. This gives main computational costs of order $\mathcal{O}(pn^2)$ for a dense sample matrix and storage costs of order $\mathcal{O}(pn)$. We used the optimized implementation of the LDA/GSVD method (Kim et al., 2005), with costs mentioned in Section 2.4. Finally, the null space method with taking advantage of the structure of covariance matrices can be implemented with order $\mathcal{O}(p^2n)$ computational costs and order $\mathcal{O}(p(p-r))$ storage costs ($p-r$ is the dimension of the null space of $\mathbf{W}$).

4. Experiments

4.1. Data description

In this section we test our algorithm on two data sets where the number of variables largely exceeds the sample size. We compare it with the methods described in Section 2. All methods were implemented with the most advantageous choice of strategies as described in Section 3.5.

The first data set is taken from the gene expression data studied in Tibshirani et al. (2002), available as `Khan` data in the `pamr` package for the `R`-environment. It consists of $n = 63$ measurements of $p = 2308$ genes belonging to $g = 4$ groups. We divided the objects by choosing randomly from every group, one half as training and one half as test set. This gave a training sample matrix of dimension 32×2308. For these data the sample matrix is dense.

The MEDLINE data (see http://www.ncbi.nlm.nih.gov/PubMed) has been used several times, among others, in the context of dimension reduction with the LDA/GSVD method (see, e.g. Howland et al., 2003; Kim et al., 2005). We use here the same data as in Kim et al. (2005), which is available at http://www-users.cs.umn.edu/~hpark/data.html. It studies the classification of documents into five groups. All groups are represented homogenously, e.g. there are 500 documents of each group. After applying a preprocessing technique we obtain $p = 22\,095$ distinct terms as explanatory variables. The corresponding object vectors have a large number of zero-entries and resulting sample matrices are sparse. We use a training set and test set with the same number of examples $n = 1250$; the number of nonzeroes of the $1250 \times 22\,095$ training sample matrix is 99 765.

Note that for both data sets the cited publications are freely available and give, among others, information on the performance of non-LDA-based methods. Thus the performance of our generalization of FLDA to the high dimension/small sample size problem can be compared with other classification methods like shrunken centroids, support vector machine, nearest neighbor methods, etc.

4.2. Results

We are here primarily interested in the costs of individual methods and in how successful they are in satisfying Fisher's criterion. We therefore compare overall time costs (measured at a server with 2 Dual Core AMD Opteron™ Processor 275 at 2191 MHz with 10 179 288 kB of memory) and the obtained between- and within-group covariance matrices. Secondarily, we add the rates of successful classification of the test data set. We used the classical and most current classification based on assigning to the class of the nearest transformed class centroid in the L_2-norm.

4.3. Gene expression data

Table 1 displays the timings of the methods we described in Section 2 and our method for the gene expression data. In the perturbation method we perturbed with the matrix $\varepsilon\mathbf{I}$ where $\varepsilon = 10^{-5}$. MP denotes the method based on the Moore–Penrose pseudo-inverse and GSVD the method from Section 2.4. In the Moore–Penrose method there was a clear gap between nonzero singular values and singular values zero to machine precision, hence the choice of a truncation parameter was trivial. Alg1 denotes our algorithm implemented as described in Section 3.4.

As we have here $p = 2308 \gg n = 63$, the acceleration with restriction to the q-dimensional complement of the common null space ($q < n$) is remarkable. The perturbation and null space methods are slow because they do not allow such a restriction. In the first case we compute $g - 1 = 3$ leading p-dimensional eigenvectors, in the second case we need at least $p - n$ eigenvectors to span the null space of $\mathbf{W}$. This explains the inferior performance of the latter method.

In Tables 2 and 3 we show the traces of between- and within-group covariance matrices from the individual methods. The null space of $\mathbf{W}$ has dimension larger than 3. Hence all methods find LDA-transformation vectors in this null space, except for the Moore–Penrose method which is defined on the complement of the null space. As for the traces of the between-group covariance matrices, we see that the criterion (11) is fully satisfied only by the perturbation and

Table 1
Gene expression data: overall computational time (in seconds) for the methods from Section 2 and our algorithm

Perturbation	MP	GSVD	Null space	Alg1
2.9	0.025	0.024	8.6	0.024

Table 2
Gene expression data: traces of between-group covariance matrices ($c^{\mathrm{T}}Bc$) achieved by the methods of Section 2 and our algorithm with growing number of transformation vectors (dimension)

Dimension	Perturbation	MP	GSVD	Null space	Alg1
1	794	183	602	794	794
2	1405	331	1148	1405	1405
3	1829	391	1715	1829	1829

Table 3
Gene expression data: traces of within-group covariance matrices ($c^T W c$) achieved by the methods of Section 2 and our algorithm with growing number of transformation vectors (dimension)

Dimension	Perturbation	MP	GSVD	Null space	Alg1
1	0	15	0	0	0
2	0	32	0	0	0
3	0	43	0	0	0

Table 4
Gene expression data: successful classification rates with L_2-norm similarity

Dimension	Perturbation (%)	MP (%)	GSVD (%)	Null space (%)	Alg1 (%)
1	74.2	51.6	51.6	74.2	74.2
2	93.6	77.4	96.8	93.6	93.6
3	96.8	83.9	96.8	96.8	96.8

null space methods and our method. The Moore–Penrose method particularly clearly fails to maximize between-group variance. In the LDA/GSVD method the failure is much less pronounced.

Table 4 displays the successful classification rates obtained with the individual methods. They correspond more or less to the relation between the traces of Tables 2 and 3. This shows that Fisher's idea to minimize within-group variance and maximize between-group variance makes sense for classifying these data.

4.4. Medline data

The dimensions for the Medline data are much larger than for the previous data as here $p = 22\,095$ and $n = 1250$. This allows us to demonstrate the benefits of our implementation compared with other fast methods that eliminate the common null space. However, we were not able to execute the perturbation and null space methods: With about $\mathcal{O}(p^2)$ storage costs we ran out of memory. For the remaining methods the overall costs, expressed by their timings, are displayed in Table 5. The table also addresses implementations of our algorithm which do not make use of the acceleration techniques from Section 3.4. The third column contains the timing for our algorithm with a direct method for the eigenproblem in step (3) and the fourth column with the product of step (2) formed as $\mathbf{B}_1 = (\mathbf{GM} - \mathbf{1_n}\bar{x}^T)((\mathbf{X}^T - \bar{x}\mathbf{1_n^T})\mathbf{V}_1\mathbf{D}_1^{-1})$. Clearly, the contribution from the issues from Section 3.4 to the high speed of our algorithm is considerable.

To explain the relatively high costs of the LDA/GSVD method we must realize that the sample matrix $\mathbf{X}$ is in this problem sparse. The LDA/GSVD method cannot profit from the sparsity; it needs the full orthogonal matrix Q of a QR-decomposition (Kim et al., 2005), giving computational costs of order $\mathcal{O}(pn^2)$. The other two methods project onto the complement of the common null space and exploit the sparsity of $\mathbf{X}$ which yields main computational costs of order $\mathsf{nnz}n$ where $\mathsf{nnz} = 99\,765$, see Section 3.4. The Moore–Penrose method is slower because it needs a full spectral decomposition of $\overline{\mathbf{W}}$ in the complement of the common null space, which has dimension $q = 1245$ for the given data.

The performance of the methods concerning approximation of Fisher's criteria can be taken from Tables 6 and 7. By definition, the Moore–Penrose method does not look for eigenvectors in the null space of $\overline{\mathbf{W}}$. This prevents the method from maximizing between-group variance in this example. The two other methods, on the contrary, first detect the two-dimensional null space and then proceed to its complement. The main difference between LDA/GSVD and our algorithm can be observed in Table 6: In the first dimension the value of $c^T B c$ is maximized only by our algorithm whereas LDA/GSVD takes "any" proper vector from the null space without taking into account $c^T B c$. However, at the second dimension (after adding the last vector from the null space of $\overline{\mathbf{W}}$) the trace of $c^T B c$ has been corrected. Table 8 displays the successful classification rates for the considered methods.

Table 5
Medline data: overall computational time (in seconds) for the Moore–Penrose method (MP), the LDA/GSVD method (GSVD) and three variants of our algorithm

MP	GSVD	Alg1, direct method	Alg1, slow product	Alg1
81	150.5	60.5	71.5	33

Table 6
Medline data: traces of between-group covariance matrices ($c^{\mathrm{T}}Bc$) achieved by the Moore–Penrose method (MP), the LDA/GSVD method (GSVD) and our algorithm with growing number of transformation vectors (dimension)

Dimension	MP	GSVD	Alg1
1	0.58	0.53	0.74
2	0.66	0.91	0.91
3	0.70	1.08	1.08
4	0.78	1.12	1.12

Table 7
Medline data: traces of within-group covariance matrices ($c^{\mathrm{T}}Wc$) achieved by the Moore–Penrose method (MP), the LDA/GSVD method (GSVD) and our algorithm with growing number of transformation vectors (dimension)

Dimension	MP	GSVD	Alg1
1	4.72e − 06	0	0
2	1.07e − 05	0	0
3	4.13e − 04	4.72e − 06	4.72e − 06
4	7.61e − 04	1.06e − 05	1.06e − 05

Table 8
Medline data: successful classification rates with L_2-norm similarity

Dimension	MP (%)	GSVD (%)	Alg1 (%)
1	51.0	31.9	48.5
2	50.2	54.6	55.0
3	63.4	74.6	74.6
4	86.7	87.5	87.5

5. Conclusions

We studied implementation of an FLDA-based classification method for the high dimension/small sample size case. We showed and confirmed with experiments that this method is closer to original FLDA than other popular FLDA-based methods. We optimized its implementation with regard to computational and storage costs using many tools, among others elimination of the common null space and sparse numerical algorithms. The resulting algorithm is prepared to be applied to very high dimensional data. It is especially fast with a sparse sample matrix. We demonstrated on examples the accelerating effect of the tools we used and we showed our implementation is faster than that of other reference methods. If the sample matrix is dense, feasibility of our algorithm depends on whether it can cope with multiplication of full $p \times n$ matrices with each other. For sample matrices that are sparse enough, the only bottleneck is the solution of a symmetric $n \times n$ eigenproblem.

Acknowledgments

We thank Prof. Eldén for turning our attention to relevant literature and we thank Alois Schloegl for his contribution in tracing an important bug in our programs. This work was supported by the Program Information Society under project 1ET400300415 (first author) and the MSMT CR Project LC536 (second author).

References

Anderson, E., Bai, Z., Bischof, C., Blackford, L.S., Demmel, J., Dongarra, J., Croz, J., Du, J., Greeenhaum, A., Hammarling, S., McKenney, A., 2000. Lapack Users' Guide. SIAM, Philadelphia.

Bai, Z., Demmel, J., Dongarra, J., Ruhe, A., van der Vorst, H. (Eds.), 2000. Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, 2000, SIAM, Philadelphia.

Bensmail, H., Celeux, G., 1996. Regularized Gaussian discriminant analysis through eigenvalue decomposition. J. Amer. Statist. Assoc. 91, 1743–1748.

Chen, L.-F., Liao, H.-Y.M., Ko, M.-T., Lin, J.-C., Yu, G.-J., 2000. A new LDA-based face recognition system which can solve the small sample size problem. Pattern Recognition 33 (10), 1713–1726.

Cheng, Y.-Q., Zhuang, Y.-M., Yang, J.-Y., 1992. Optimal Fisher discriminant analysis using the rank decomposition. Pattern Recognition 25 (1), 101–111.

Duda, R.O., Hart, P.E., Stork, D.G., 2000. Pattern Recognition. second ed. Wiley, New York.

Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Ann. Eugenics 7, 179–188.

Friedman, J.H., 1989. Regularized discriminant analysis. J. Amer. Statist. Assoc. 84, 165–175.

Golub, G.H., van Loan, C.F., 1996. Matrix Computations. third ed. John Hopkins University Press, Baltimore.

Guo, Y.-F., Li, S.-J., Yang, J.-Y., Shu, T.-T., Wu, L.-D., 2003. A generalized Foley–Sammon transform based on generalized Fisher discriminant criterion and its application to face recognition. Pattern Recognition Lett. 24, 147–158.

Hastie, T., Tibshirani, R., 2003. Expression arrays and the $p \gg n$ problem. See ⟨http://www-stat.stanford.edu/∼hastie/Papers/pgtn.pdf⟩.

Hoffbeck, J.P., Landgrebe, D.A., 1996. Covariance matrix estimation and classification with limited training data. IEEE Trans. Pattern Anal. Mach. Intell. 18 (7), 763–767.

Hong, Z.-Q., Yang, J.-Y., 1991. Optimal discriminant plane for a small number of samples and design method of classifier on the plane. Pattern Recognition 24 (4), 317–324.

Howland, P., Park, H., 2004. Generalizing discriminant analysis using the generalized singular value decomposition. IEEE Trans. Pattern Anal. Mach. Intell. 26 (8), 995–1006.

Howland, P., Jeon, M., Park, H., 2003. Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition. SIAM J. Matrix Anal. Appl. 25 (1), 165–179.

Howland, P., Wang, J., Park, H., 2006. Solving the small sample size problem in face recognition using generalized discriminant analysis. Pattern Recognition 39, 277–287.

Johnson, R.A., Wichern, D.W., 1998. Applied Multivariate Statistical Analysis. Prentice-Hall, Engelwood Cliffs, NJ.

Kim, H., Howland, P., Park, P., 2005. Dimension reduction in text classification with support vector machines. J. Mach. Learn. Res. 6, 37–53.

Krzanowski, W.J., Jonathan, P., McCarthy, W.V., Thomas, M.R., 1995. Discriminant analysis with singular covariance matrices: methods and applications to spectroscopic data. Appl. Statist. 44 (1), 101–115.

Li, Y., Kittler, J., Matas, J., 1999. Effective implementation of linear discriminant analysis for face recognition and verification. In: Leonardis, A., Solina, F. (Eds.), Lecture Notes in Computer Science 1689. Springer, Berlin, pp. 234–242.

MathWorks, Inc., 1984–2005. MATLAB 7.0, ⟨http://www.mathworks.com/products/matlab/⟩.

Moler, C.B., Stewart, G.W., 1973. An algorithm for generalized matrix eigenvalue problems. SIAM J. Numer. Anal. 10 (2), 241–256.

Paige, C.C., Saunders, M.A., 1981. Towards a generalized singular value decomposition. SIAM J. Numer. Anal. 18 (3), 398–405.

Parlett, B.N., 1998. The Symmetric Eigenvalue Problem. SIAM, Philadelphia.

R Development Core Team, 2005. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna.

Ripley, B.D., 1996. Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge.

Tibshirani, R., Hastie, T., Narasimhan, B., Chu, G., 2002. Diagnosis of multiple cancer types by shrunken centroids of gene expression. Proc. Nat. Acad. Sci. USA 99 (10), 6567–6572.

Venables, W.N., Ripley, B.D., 2002. Modern Applied Statistics with S. fourth ed. Springer, New York.

Yang, J., Yang, J.-Y., 2003. Why can LDA be performed in PCA transformed space? Pattern Recognition 36 (2), 563–566.

Yang, J., Yu, H., Kunz, W., 2000. An efficient LDA algorithm for face recognition. Sixth International Conference on Control, Automation, Robotics and Vision (ICARCV2000).