EXTENDED KRYLOV SUBSPACE FOR PARAMETER DEPENDENT SYSTEMS *

V. SIMONCINI†

Abstract. The Extended Krylov Subspace has recently received considerable attention as a powerful tool for matrix function evaluations and other problems involving large matrices. In this paper we show that this space has a great potential within projection-type methods for effectively solving several other important large-scale algebraic problems: we focus on the solution of shifted systems and of more general parameter-dependent matrix equations, and on the approximation of the transfer function by projection. Numerical experiments stemming from real applications show the effectiveness of the approach.

1. Introduction. The Extended Krylov subspace has recently emerged as a competitive tool for approximating the solution to various problems involving large matrices. Introduced by Druskin and Knizhnerman in [10] for approximating functions of symmetric matrices, it was later explored to approximate functions of general non-symmetric matrices [24], and to approximate the solution to the Lyapunov, Sylvester and Riccati equations by means of projection-type methods [20, 21, 35].

Given a nonsingular matrix $A$ and a vector $b$, the Extended Krylov subspace provides an enhanced approximation space by using both powers of $A$ and $A^{-1}$, namely

$$
EK_m(A, b) = \text{span}\{b, A^{-1}b, Ab, A^{-2}b, A^2b, \ldots, A^{m-1}b, A^{-m}b\}
= K_m(A, b) + K_m(A^{-1}, A^{-1}b),
$$

where $K_m(A, b) = \text{span}\{b, Ab, \ldots, A^{m-1}b\}$ is the Standard Krylov subspace. Clearly, $\dim(EK_m(A, b)) \leq 2m$. It was shown in [35] that the space $EK_m(A, b)$ and the matrix representing $A$ in that space may be generated iteratively, so that $EK_m(A, b) \subseteq EK_{m+1}(A, b)$. This important property makes it possible to implement an iterative method based on the Extended Krylov subspace by means of a simple recurrence.

Projection-type methods determine an approximation to the sought after solution by projecting the given problem onto a possibly much smaller approximation space, and in case of equations, by then enforcing some condition to obtain a unique approximation in the given space. Approximation spaces that have been extensively explored in the past in a variety of problems are the standard Krylov space, the shift-invert Krylov space, and the rational Krylov space. The last two spaces have the shortcoming of requiring a possibly expensive parameter selection, and their performance may in some cases be very sensitive to an inaccurate selection of these parameters. On the other hand, neither the Standard nor the recently introduced Extended Krylov subspaces depend on parameters. In addition, the Extended Krylov subspace has the advantage of “approximating both ends of the spectrum”, by generating powers of $A$ and of $A^{-1}$, and their combination may intuitively explain its effectiveness over using either powers of $A$ (standard Krylov space) or of $A^{-1}$ (invert Krylov space). However, a rigorous understanding of the enhancements obtained by the Extended Krylov space is far less intuitive. A convergence analysis for approximating matrix functions for a large class of functions can be found in [24] (see also [10] for the symmetric case); a convergence analysis for solving the Lyapunov equation can instead be found in [26] for $A$ symmetric, while it is under investigation in [25] for general $A$.

*Version of 20 November 2009.
†Dipartimento di Matematica, Università di Bologna, Piazza di Porta S. Donato 5, I-40127 Bologna, Italy and CIRSA, Ravenna, Italy (valeria@dm.unibo.it).
In this paper we show that the Extended Krylov subspace has a great potential for effectively solving several important large-scale algebraic problems that can be found in many applied fields. Here we focus on the solution of shifted systems and of more general parameter-dependent matrix equations, and on the approximation of the transfer function. Our numerical results show that the new proposed projection methods based on the Extended Krylov subspace significantly outperform state-of-the-art methods.

The solution of parameter dependent linear systems of the form

\[(A + \sigma E)x = b,\]  

where \(A, E \in \mathbb{C}^{n \times n}, b \in \mathbb{C}^n, \sigma \in \mathbb{C}\) and \(x = x(\sigma)\), arises in many applications such as control theory, structural dynamics and time-dependent PDEs; see, e.g., [11, 29, 36] and references therein. Depending on the application problem, the values of the parameter \(\sigma\) may vary either in a neighborhood of a given fixed quantity \(\sigma_0\), or in a wide range. In the latter case, strategies based on perturbation methods around \(A + \sigma_0 E\) are clearly not applicable. It is important to realize that for shifted systems, the computational challenge is given both by the properties and dimensions of \(A\) and \(E\), but also by the number of shifts involved. Even for medium size systems, for which sparse direct methods are usually very efficient, the presence of many parameter values makes it necessary to determine strategies whose cost grows only sub-linearly with the number of parameters. We refer to, e.g., [4, 5, 12, 13, 34] for some solvers and preconditioners that take into account these properties; a more complete account may be found in [36].

As a natural extension to the problem (1.1), the right-hand side \(b\) could also depend on the parameter, namely \(b = b(\sigma)\) (see, e.g., [7, 17, 27]), so that for a set of values \(\sigma_1, \ldots, \sigma_s\), the following problem becomes of interest

\[AX + EXS = B, \quad S = \text{diag}(\sigma_1, \ldots, \sigma_s), \quad B = [b(\sigma_1), \ldots, b(\sigma_s)].\]

Another closely related problem with wide application in Control consists of approximating the so-called transfer function

\[h(\sigma) = C^*(A - i\sigma E)^{-1}B, \quad \sigma \in \mathbb{R},\]

where \(A, E, B\) and \(C\) are the matrices representing a dynamical linear system. Here and throughout the paper we use \(x^\top\) to denote transposition of a complex vector \(x\), and \(x^*\) to denote conjugate transposition. The transfer function may have a quite oscillatory behavior on a range of values for \(\sigma\) that usually spans several orders of magnitude, and such behavior needs to be captured by a reliable approximation.

In the following sections we first characterize the Extended Krylov subspace, describing in some detail its computation. We then focus on the three problems above, providing for each of them relevant numerical experiments with data stemming from real application problems, that emphasize the enhancement obtained by employing the new approximation space. All reported experiments were run using Matlab 7.5 [28].

2. The Extended Krylov subspace. An orthogonal basis of the Extended Krylov subspace may be generated iteratively while the space is expanded by means of a Gram-Schmidt procedure, in a way similar to that of the standard Krylov subspace. A less straightforward procedure needs to be devised to iteratively generate the projection and restriction matrix that represents \(A\) onto the generated space,
without additional matrix-vector multiplies with $A$. More precisely, if the (orthonormal) columns of $V_m \in \mathbb{C}^{n \times 2m}$ span $\mathcal{EK}_m(A, b)$, then an ad-hoc procedure is required to obtain $T_m = V_m^* A V_m$, since the Gram-Schmidt procedure delivers matrix-vector products with $A$ for free only for the odd columns of $V_m$. In [35] a recurrence was suggested to obtain $T_m$: The odd columns coincide with the corresponding columns of the orthogonalization coefficients matrix in the Gram-Schmidt process, whereas the even columns are recovered by means of an auxiliary recurrence with short-length vectors [35, Prop. 3.2].

The generation of $V_m = [V_1, V_2, \ldots, V_m] \in \mathbb{C}^{n \times 2m}$ and $T_m = (t_{i,j}) \in \mathbb{C}^{2m \times 2m}$ may be summarized in the following steps:

**Algorithm 1.**

Set $V_1 = V_1 = \text{gram}(b, A^{-1} b)$.

For $m = 1, 2, \ldots$,

1. Set $V_m^{(1)}$: first column of $V_m$; $V_m^{(2)}$: second column of $V_m$.
2. $V_m' = [AV_m^{(1)}, A^{-1} V_m^{(2)}]$.
3. $\hat{V}_{m+1} \leftarrow$ orthogonalize $V_m'$ w.r.to $V_m$;
4. $V_{m+1} = \text{gram}(\hat{V}_{m+1})$.
5. Let $h$ (size $(2m+2) \times 2$) contain the orth. coeffs.
6. New odd column of $T_m$: $t_{2m-1} = h_{1,1}$.
7. New even column of $T_m$:

   - $(m = 1)$ $t_{1,2} = h_{1,1}(\ell_{11}^{(1)}) - 1 e_1^{(1)}\ell_{22}^{(1)}$;
   - $t_{1,4} = e_2 - T_m h_{1,2,2} \ell_{22}^{(2)}$.
   - $(m > 1)$ $t_{2m+1} = t_{2m} + t_{2m-1} \rho^{(m)}$,$\rho^{(m+1)} = (h_{11}^{(m+1)}) - \ell_{12}^{(m+1)}$.

8. $V_{m+1} = [V_m, V_{m+1}]$.

Algorithm 1 adds two new vectors to the basis to expand the space, one multiplied by $A$ and the other by $A^{-1}$. The function $\text{gram}$ performs a Gram-Schmidt orthogonalization within the matrix block.

A closer look at the relations generating $T_m$ shows that $T_m$ is block upper Hessenberg, with $2 \times 2$ blocks ($2p \times 2p$ if $b$ is a matrix with $p$ columns, see below). In exact arithmetic it is block tridiagonal and Hermitian whenever $A$ is Hermitian; see also [23] for a different derivation for $A$ real symmetric. As soon as the representation matrix $T_m$ is updated, the underlying problem may be approximated within the enhanced space, and then the approximate solution prolonged back to $\mathbb{C}^{n}$; see, e.g., Algorithm 2 below.

For later convenience, we remark that a block version of the space, namely $\mathcal{EK}_m(A, B)$, with $B \in \mathbb{C}^{n \times p}$, can also be generated, requiring only a “blocking” of the scalar quantities described above, and no other major changes [35].

We also mention that the matrix $V_m$ formally satisfies the following Arnoldi-type relation (see, e.g., [10, 21]):

$$AV_m = V_m T_m + V_{m+1} \tau_{m+1, m} E_m^T,$$

where $E_m$ contains the last 2 columns of the identity matrix of dimension $2m$, and $\tau_{m+1, m} = V_{m+1}^* AV_m$; note that the second row of $\tau_{m+1, m}$ is zero [35]. Clearly, the relation in (2.1) cannot be used to explicitly derive the next column vectors in $V_{m+1}$.
For a vector, it was shown in [24, 35] that the method may break down, but then it is a happy breakdown, that is, the problem solution is contained in the computed approximation space. For \( b = B \) a tall matrix, then unlucky breakdown may occur, and this may be due to the redundancy of some of the columns used to enlarge the space at any iteration. Deflation techniques should then be adopted, as it is the case in block Krylov subspace methods; see [18] for a rather complete list of references on the topic. The special structure of the basis makes the deflation implementation more cumbersome. We refer to [37] for an implementation.

Whenever the matrix \( A \) is complex symmetric, namely \( A = A^\top \in \mathbb{C}^{n \times n} \), it is still possible to exploit this form of symmetry, to obtain a block tridiagonal matrix, as in the real symmetric case. Following the procedure employed for standard Krylov subspace methods, this may be obtained by using the bilinear form \( (x, y) = x^\top y \), with \( x, y \in \mathbb{C}^n \) throughout the algorithm, instead of the standard Euclidean inner product \( x^* y \), where \( x^* \) denotes the conjugate transpose of \( x \); we refer to [36] for a description of the use of the bilinear form \((x, y)\) in the complex symmetric context. As a result, the matrix \( V_m \) is complex and satisfies

\[
V_m^\top V_m = I_{2p_m},
\]

where \( p_m \) is the number of columns of \( b \). Moreover, as anticipated, \( T_m \) is block tridiagonal with \( 2p \times 2p \) blocks, and as in the real symmetric case no explicit orthogonalization is required with respect to the older basis vectors.

3. Application problems. We next describe some application problems where the use of the Extended Krylov subspace indeed significantly improves convergence, with respect to the most direct competitors. The fact that each iteration requires the application of \( A^{-1} \) readily shows that for the same space dimension, the approximation enhancement obtained by using the new space should be sufficiently high to make up for the extra cost of using \( A^{-1} \), in some forms, rather than merely \( A \). We show that this is indeed the case for many applications we have encountered.

3.1. Shifted linear systems. We consider the solution of the parameterized linear system

\[
(A + \sigma I)x = b, \quad \sigma \in \mathbb{C}
\]

which need be solved for many values of \( \sigma \). The solution \( x = x(\sigma) \) may be written as

\[
x = (A + \sigma I)^{-1}b \equiv f(A)b.
\]

Therefore, this is a special case of matrix function evaluation, and the Extended Krylov subspace can be employed as done for instance in [10, 24]. However, the linear system setting is particularly convenient to derive several useful properties for shifted systems (see, e.g., [36, sec.14]). In particular, using (2.1), it follows that

\[
(A + \sigma I)V_m = V_m(T_m + \sigma I) + V_{m+1} \tau_{m+1,m} E_m^\top,
\]

showing that the same space \( \text{EK}_m(A, b) \) can be used to approximate all parameterized systems, as \( \sigma \) varies, the way it is done for the usual space \( K_m(A, b) \) [36, sec.14]. Therefore, using the fact that \( V_m^* b = e_1 \| b \| \), we have

\[
x_m(\sigma) = V_m y_m(\sigma) := V_m(T_m + \sigma I)^{-1} e_1 \| b \|.
\]

Clearly, the parameter \( \sigma \) only enters the computation in the evaluation of \( y_m(\sigma) \). Note that, for each \( \sigma \), this procedure corresponds to the interpolation approximation of \( f \) as \( y_m(\sigma) = f(T_m) e_1 \| b \| \) [24].

The major difference with respect to the standard Krylov approach is that a system with \( A \) must be solved at each iteration to expand the basis, making the
generation of the single space more expensive. The computation of $\text{EK}_m(A, b)$ may be advantageous with respect to solving each system $(A + \sigma I)$ if either of the following conditions applies:

i) Solving with $(A + \sigma I)$ is more expensive than solving with $A$ (e.g., $A$ is large and real and $\sigma$ is complex);

ii) The number of parameters to be considered is large, and the space dimension at convergence is significantly lower.

Both conditions can be satisfied in application problems. In particular, below we consider an example from structural dynamics that does require the solution of a large number of parameterized systems. We also remark that shifted systems are difficult to precondition, while maintaining the shifted structure (cf. the discussion in [36]), therefore the new enhanced space may be viewed as a form of acceleration alternative to spectral transformations.

Whenever $A$ is symmetric, it is known that the solution of a linear system by a Krylov subspace does not require storing and explicitly orthogonalizing the whole basis. As already mentioned, this is also the case for the Extended Krylov subspace: since $T_m$ is block tridiagonal, only a three-term block recurrence needs to be performed. More precisely, the orthogonalization in step 4 of Algorithm 1 need be carried out only with respect to the blocks $V_m , V_{m - 1}$. We refer to [23] for additional details of an explicit implementation of this short-term recurrence.

We next give a possible outline of the resulting iterative scheme.

**Algorithm 2.**

Given $A, r_0(\sigma) = b, x_m(\sigma) = 0, \Sigma$.

Compute $\beta_0(\sigma) = \|b\|, v_1 = b/\beta_0(\sigma)$. Set $\Sigma_c = \emptyset$.

Compute $V_1 = V_1 = \text{gram}(b, A^{-1}b)$.

for $m = 1, 2, \ldots$

1. Generate $V_{m+1}$, compute new columns of $T_m$;

2. Solve $(T_m + \sigma I)y(\sigma) = e_1\|b\|, \sigma \in \Sigma \setminus \Sigma_c$;

3. Update set $\Sigma_c$ of converged systems;

4. if $\Sigma \setminus \Sigma_c = \emptyset$ compute $x_m(\sigma) = V_m y_m(\sigma), \sigma \in \Sigma_c$; stop;

5. Expand space: $V_{m+1} = [V_m, V_{m+1}]$.

The Arnoldi-type relation (2.1) can also be used to generalize other convenient properties. Most importantly, the following result holds for the system residuals.

**Proposition 3.1.** Let $r_m(\sigma) = b - (A + \sigma I)x_m(\sigma)$, with $x_m$ as in (3.1). Then $r_m(\sigma) = -V_{m+1}^T \tau_{m+1, m} E_m y_m(\sigma)$. In particular, all residuals are collinear to $V_{m+1} e_1$.

**Proof.** The expression for the residual is obtained by inserting the Arnoldi-type relation (2.1) into $r_m(\sigma) = b - (A + \sigma I) V_m y_m(\sigma)$. The collinearity follows from the normality of the basis and from noticing that the second row of $\tau_{m+1, m}$ is zero. □

This property was first exploited for restarting shifted system solvers in [34]; we also refer to [14] for error estimations of Krylov subspace solvers for shifted systems that make use of such a collinearity property.

Two important features follow from Proposition 3.1. The first one is that the residual norm can be computed cheaply at each iteration for all shifted systems, without the explicit computation of the solution $x_m(\sigma)$ or the residual, namely

$$\|r_m(\sigma)\| = \|\tau_{m+1, m} E_m^T y_m(\sigma)\| = \|e_1^T \tau_{m+1, m} E_m^T y_m(\sigma)\|,$$  \hspace{1cm} (3.2)
where the second equality follows from the fact that the second row of $\tau_{m+1,m}$ is zero (cf. section 2).

The second consequence is of paramount importance when solving nonsymmetric systems, for which restarting may be required. Since all basis vectors need be explicitly orthogonalized and stored, a maximum subspace dimension is usually allowed, and if accuracy is not satisfactory, the procedure is restarted with the current approximate solution as starting guess, so that the new space is generated with the current residual as starting vector. When dealing with shifted systems, the fact that the residuals are collinear makes it possible to build a single approximation space at restart time, starting with $V_{m+1}e_1$; otherwise, a distinct restart space would be required for each parameterized system. We refer to [35, 36] for a discussion on general Galerkin approximations and their collinearity properties.

The restarted process can be outlined as follows:

Given $A, r_0(\sigma) = b, x_m(\sigma) = 0, \Sigma, m$.

Compute $\beta_0(\sigma) = \|b\|, v_1 = b/\beta_0(\sigma)$. Set $\Sigma_c = \emptyset$.

1. Build $EK_m(A, v_1)$ as in Algorithm 1;
2. Compute $y_m(\sigma) = (T_m + \sigma I)^{-1}e_1 \beta_0(\sigma), \sigma \in \Sigma \setminus \Sigma_c$;
3. Compute $\|r_m(\sigma)\|$ using (3.2)
4. Compute $x_m(\sigma) = x_m(\sigma) + V_m y_m(\sigma), \sigma \in \Sigma \setminus \Sigma_c$. Update set $\Sigma_c$ of converged shifted systems. If $\Sigma \setminus \Sigma_c = \emptyset$ stop;
5. Set $v_1 = V_{m+1}e_1$. Set $\beta_0(\sigma) = v_1^* r_m(\sigma), \sigma \in \Sigma \setminus \Sigma_c$. Goto 1.

3.2. Numerical Experiments. In this section we describe some problems that can benefit from the use of the shifted Extended Krylov subspace. Comparisons are performed with respect to the most direct competitor, shifted FOM [34]. Experiments are also reported in some cases with a sparse direct solver applied to each shifted system. In all cases, prior to the generation of the space $EK_m(A, b)$, the matrix $A$ is reordered (and reordering in Matlab) and then factorized once for all.

Example 3.2. We consider the nonsymmetric matrix $A$ stemming from the centered finite difference discretization of the operator $L(u) = -\Delta u + 50(x+y)(u_x + u_y)$ in the unit square, with Dirichlet homogeneous boundary conditions. The matrix $A$ is scaled by the mesh dimension, and we consider three mesh refinements, resulting into a problem of dimension $n = 2500, 10000$ and $n = 160000$, respectively. The right-hand side is chosen as the normalized vector of all ones, and the shifts are taken to be 50 values uniformly distributed in the interval $[0, 5]$. The stopping tolerance is set equal to $10^{-8}$. The results using the shifted versions of restarted FOM and of restarted EKSM are reported in Table 3.1, and show that the Extended space is able to effectively reduce the number of restarts and the CPU time. The cost of factorizing the coefficient matrix is included. For the sake of completeness, we also mention that full GMRES on the largest system with $\sigma = 0$ showed a residual norm of the order of 0.1 after 500 iterations and 2160 seconds of CPU time. In addition, the Matlab “backslash” (system-solve) operation required 2.52 secs to solve a single reordered system of size 160000, for a total of 1260 secs to solve all 500 systems.

Example 3.3. We consider the rational approximation of the action of a matrix function to a vector. More precisely, we compute the partial fraction sum

$$\sum_{k=1}^{\nu} \kappa_k (\omega_k^2 I - A)^{-1} A b \approx f(A) b$$
Table 3.1

<table>
<thead>
<tr>
<th>$n_{\text{subspace}}$</th>
<th>$\text{FOM}$</th>
<th>$\text{EKSM}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2500$</td>
<td>$10$</td>
<td>$1.50$ (41)</td>
</tr>
<tr>
<td>$2500$</td>
<td>$20$</td>
<td>$1.57$ (14)</td>
</tr>
<tr>
<td>$10,000$</td>
<td>$10$</td>
<td>$3.23$ (49)</td>
</tr>
<tr>
<td>$10,000$</td>
<td>$20$</td>
<td>$4.35$ (27)</td>
</tr>
<tr>
<td>$160,000$</td>
<td>$10$</td>
<td>$399.80$ ($&gt;300$)</td>
</tr>
<tr>
<td>$160,000$</td>
<td>$20$</td>
<td>$356.63$ (97)</td>
</tr>
</tbody>
</table>

Example 3.2. Shifted solvers for nonsymmetric matrices of size $n$. 500 Shifts uniformly distributed in $[0, 5]$. Stopping tolerance $10^{-8}$.

where $f(\lambda) = \sqrt{\lambda}$. We test the method on the matrix $A \in \mathbb{R}^{2500 \times 2500}$, discretization of the operator $-\Delta u + 10u_x$ on the unit square with homogeneous Dirichlet boundary conditions; $b$ is the vector of all ones. Here we employ the rational approximation described as method2 in [19, p.2512-2513], which yields complex values of the shifts $\omega_2^2$’s. We use $\nu = 18$ (non-conjugate) terms which in our setting provided a relative error in the function evaluation below $10^{-13}$. On the other hand, this requires solving 18 shifted linear systems: we used a stopping tolerance equal to $10^{-10}$. A small matrix is considered to be able to use method2 exactly, without any estimate of the involved quantities. The Extended Krylov subspace method took 0.04 secs to factorize the matrix, and 0.33 secs to solve all systems. We recall here that most computation is carried out in real arithmetic, and only the solution of the small linear system in $T_k - \omega_2^2 I$ and the solution update are performed in complex arithmetic. The sequential solution of all (complex non-Hermitian) systems by a sparse direct method required about 1.28 secs.

Example 3.4. We consider a benchmark test problem stemming from a structural model of a car windscreen [8]. The system to be solved is written as

$$(K - \frac{\omega^2}{\tau}M)x(\omega) = b,$$

and has size $n = 22,692$. Here $K$ and $M$ are the positive semi-definite stiffness and positive definite mass matrices, respectively, while $\tau = 1 + i\gamma$, where $\gamma = 0.1$ is the chosen value for natural damping. The frequency range for $\omega$ is $[0.5, 200]$. We focus on 11 values uniformly distributed in the subinterval $[185, 200]$. Since $K$ is singular, and to exploit the shifted form, we rewrite the equation as

$$\left(M^{-1}(K - \frac{\omega^2}{\tau}M) + \frac{\omega^2 - \omega_0^2}{\tau}I\right)x(\omega) = M^{-1}b$$

Note in particular the left multiplication by $M^{-1}$, which is cheaper than right multiplication, since in the latter case the post-processing for recovering $x(\omega)$ would involve a solve with $M$ for each value of $\omega$. Here $\omega_0$ is chosen so that $K - \frac{\omega^2}{\tau}M$ is nonsingular, and more precisely $\omega_0$ is the real part of the 5th value among the considered shifts. We compare the performance of the Extended Krylov method with respect to shifted Lanczos in the $M$-inner product, left preconditioned by $K - \frac{\omega^2}{\tau}M$, with $\omega = -100$ as suggested in [29]; see also [30]. Full reorthogonalization of the basis vectors is performed. We recall that this type of preconditioning corresponds to a spectral transformation, and that the value of $\omega$ was specifically selected in [29] to
Table 3.2

Example 3.4. Performance of methods. Reported are the best timings among several runs.

<table>
<thead>
<tr>
<th></th>
<th>Extended Krylov</th>
<th>Shift-Invert Arnoldi</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU time (secs)</td>
<td>126.06</td>
<td>395.22</td>
</tr>
<tr>
<td>Space dimension</td>
<td>24</td>
<td>36</td>
</tr>
<tr>
<td>Number of $K - \frac{\alpha}{\tau}M$ solves</td>
<td>12</td>
<td>36</td>
</tr>
</tbody>
</table>

improve the spectrum of the transformed problem. We also recall that the matrix-vector multiplications can be performed so as to exploit the following relation:

$$(K - \frac{\hat{\omega}^2}{\tau} M)^{-1}(K - \frac{\omega^2}{\tau} M) = I + \frac{\omega^2 - \hat{\omega}^2}{\tau} (K - \frac{\hat{\omega}^2}{\tau} M)^{-1} M,$$

so that a single Krylov subspace with the matrix $(K - \frac{\hat{\omega}^2}{\tau} M)^{-1} M$ needs to be generated for all shifts.

Table 3.2 reports the performance of the two methods on this problem to achieve a final relative (preconditioned) residual norm less than $10^{-8}$, with the understanding that the final error norm may significantly differ for the two methods. Noticing that a single system solve with the matrix $K - \alpha \tau M$, $\alpha = \omega_0, \hat{\omega}$ takes about 8.5 seconds, it is clear that most computational time is spent in these matrix-vector multiplies. We should mention that the new method also requires solving with $M$, but these appear to be much less expensive (0.1 secs on average). The advantages of the new approach are clearly visible. Both methods require factorizing some (reordered) matrices at the beginning; these costs were not included in the total count. As concluding computational remark, we report that swapping was observed in many of the performed runs, especially for the Shifted-Inverted Arnoldi method, thus the best timings were considered. In particular, the subspace dimension influences the memory requirements for storing the space basis, in both methods.

4. Sylvester-type equations. In some applications the right-hand side may also depend on the parameter. In this case, considering the parameters $\sigma_1, \ldots, \sigma_s$, the systems to be solved take the form

$$AX + XS = B, \quad S = \text{diag}(\sigma_1, \ldots, \sigma_s), \quad B = [b(\sigma_1), \ldots, b(\sigma_s)].$$

This is a particular form of Sylvester equation, where in general $S$ may be nondiagonal. To ensure uniqueness of the solution, $A$ and $-S$ must have disjoint spectra. In fact, we could formulate the problem for any matrix $S$ not necessarily diagonal. Note that the more general equation $AX + ESX = B$ may be restated as the one above, as long as $E$ is nonsingular (see also below).

For $s \ll n$, an approximation to $X$ may be effectively obtained in an approximation space generated by using $A$ and $B$; see, e.g., [6, 9, 20, 22, 31, 33]. In the Extended Krylov subspace context, we can generate the matrices $V_m, T_m$ by means of Algorithm 1, and then, following the Galerkin requirements, project the problem onto the space and require that the residual be orthogonal to it. More precisely, we can write $X \approx V_m Y$, for some $Y$ such that

$$V_m^* R_m \equiv V_m^* (AV_m Y + V_m YS - V_m E_1 \beta_0) = 0,$$

where $B = V_m E_1 \beta_0$, $\beta_0 \in \mathbb{C}^{s \times s}$, which is equivalent to

$$T_m Y + YS = E_1 \beta_0. \quad (4.1)$$
The Extended Krylov subspace for parameter dependent systems

Here and in the following, $E_1$ denotes the first $s$ columns of the identity matrix, whose dimension is made clear from the context. Symmetry (real or complex) can be exploited as described in section 2, lowering the computational cost of the method. This formulation may be viewed as a special case of the generalization proposed in [20], where $S$ has small dimension and the whole matrix $B$ is used. In fact, since $S$ is still diagonal, the equation (4.1) still consists of $s$ separate small shifted linear systems, therefore they can be solved very efficiently.

As in the previous framework, the residual norm may be computed inexpensively as $\|R_m\| = \|e_1^T \tau_{m+1,m} E_m^* Y_m(\sigma)\|$, where $Y_m$ solves the reduced problem (4.1). The only difference with respect to the (scalar) shifted system is that the right-hand side $E_1 \beta_0$ is a full rank matrix, whereas in the former case, the right-hand side is the rank-one matrix $e_1\|b\| \mathbf{1}^T$, that is a replication of the same vector $e_1\|b\|; T_m y + y\sigma_1 = e_1\|b\|, \ldots, T_m y + y\sigma_s = e_1\|b\|$.

We explicitly remark that the approach generates the block Extended Krylov subspace $\mathbf{E}K_m(A,B)$, whose dimension is less or equal to $2ms$, where $s$ is the number of columns of $B$. Therefore the method seems to be appealing when $s$ is moderate, to avoid the presence of very large blocks. A particularly attractive situation occurs when $B$ is rank deficient. In such a setting, $B = \mathcal{V}_m E_1 \beta_0$, with $E_1 \in \mathbb{R}^{2s \times m \times s}$, $\beta_0 \in \mathbb{C}^{s \times s}$, and $s_0 = \text{rank}(B)$. Therefore, the Extended space is generated using $s_0 \leq s$ vectors; we discuss such an example next.

**Example 4.1.** This example deals with the solution of parameterized algebraic systems in a direct frequency analysis arising in structural dynamics [11]. The general frequency dependent equation of motion is given by

$$(K^* + i\sigma C_V - \sigma^2 M)x(\sigma) = b(\sigma). \tag{4.2}$$

Here we consider no viscous damping, so that $C_V = 0$. The remaining data matrices are the stiffness and mass matrices of a finite element discretization of a machine foundation resting on ground (Case B in [11]), and have size $3627 \times 3627$. The stiffness matrix has in fact the form $K^* = K + iC_D$, where $K$ is the true stiffness matrix and $C_D$ accounts for hysteretic damping, so that $K^*$ is complex symmetric. We consider $s$ frequencies $f_k$ uniformly taken in the interval $[10, 20] \text{Hz}$, and $\sigma_k = 2\pi f_k, k = 1, \ldots, s$. Following [17], the right-hand side is given as $B = F \Phi(\sigma)$, with $F = [e_k, \ldots, e_k]$ and $\Phi_{i,j} = \exp(-\sigma_j(i-1)0.4\pi), i = 1, \ldots, c, j = 1, \ldots, s$. For these tests, we have chosen $c = 2$ and $k_1 = 1962, k_2 = 2046$. Note that $B = [b(\sigma_1), \ldots, b(\sigma_s)]$ is highly rank deficient, as it has rank at most $c$, whereas the number $s$ of columns may be much larger. It is important to realize that if we let the columns of $F_0$ span the space of $F\Phi$, then the Extended Krylov subspace is generated with starting matrix $F_0$, thus significantly reducing the computational costs in case $\text{rank}(F_0) \leq c \ll s$. The systems $(K^* - \sigma^2 M)x(\sigma_k) = b(\sigma_k), k = 1, \ldots, s$ can be written in compact form and with obvious notation as

$$K^* X + M X S = B, \quad S = \text{diag}(\sigma_1^2, \ldots, \sigma_s^2).$$

For $M$ nonsingular, let $M = L U$ be its LU decomposition. Then we can solve the following equivalent Sylvester equation

$$(L^{-1} K^* U^{-1}) \tilde{X} + \tilde{X} S = L^{-1} B, \quad \tilde{X} = U X. \tag{4.3}$$

Note that this formulation allows the iterative solver and the computation of the residual norm to be less sensitive to the conditioning of $M$, compared to the left-side inversion. It also allows one to exploit the symmetry of $M$, as discussed next.
We should mention that in many applications $M$ is much sparser than $K^*$ (e.g., the possibly diagonal mass matrix), therefore its LU decomposition is not computationally expensive. Otherwise, one can consider right or left inversion, and replace the LU decomposition with an iterative solution with $M$.

The equation in (4.3) is solved by projection onto the Extended Krylov subspace. In the considered problem, $M$ is real symmetric, so that $L = U^\top$, and $K^*$ is complex symmetric, so that $L^{-1}K^*U^{-1}$ is also complex symmetric. To fully exploit the symmetry of the problem, we have implemented the Extended Krylov method with the bilinear form $(x, y) = x^\top y$ instead of the standard complex inner product $x^*y$; see, e.g., [36]. It follows that $\tau$-orthogonalization is only carried out with respect to the previous 2 block vectors (cf. section 2). Systems with $K^*$ are solved either with a sparse direct solver, or with a complex symmetric CG method with LDL$^\top$-type preconditioning (see below). Note that the sparse factors (in either the direct solver or in the preconditioner) are computed once for all before the procedure is started. Their cost is included in the reported timings.

We compare the performance of the new solver with the solution of the original equation $Q(\sigma)x(\sigma) = b(\sigma), Q(\sigma) = K^* - \sigma^2 kM$ for each $\sigma$, by means of the complex symmetric CG method, preconditioned by an LDL$^\top$-type preconditioner with dropping tolerance $10^{-2}$ (GMRES on each system $Q(\sigma)x(\sigma) = b(\sigma)$ was significantly more expensive for the same number of iterations). The CPU Time for building each new preconditioner is included in the total computational cost. For all iterative solvers, the stopping tolerance was set equal to $10^{-10}$. For such a small problem size, we expect a sparse direct solver to perform considerably well, compared to iterative methods when solving $Q(\sigma)x(\sigma) = b(\sigma)$. For the sake of completeness we report the performance of the effective Matlab sparse “backslash” solver, showing that the presence of many shifts makes the approach less suitable, since the computational costs grows linearly with the number of shifts. We expect a much more pronounced difference in favor of the iterative solvers for larger problems.

<table>
<thead>
<tr>
<th>s</th>
<th>Sparse Direct Solver</th>
<th>Complex sym CG</th>
<th>Extended Krylov Direct solves $K^*$</th>
<th>Extended Krylov Iter.solves $K^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>12.57</td>
<td>48.30 (108)</td>
<td>1.59 (36)</td>
<td>8.00 (36)</td>
</tr>
<tr>
<td>100</td>
<td>24.88</td>
<td>96.33 (108)</td>
<td>2.12 (40)</td>
<td>9.69 (40)</td>
</tr>
</tbody>
</table>

The reported CPU Times (in seconds) show the effectiveness of the new approach, with either direct or iterative solves with $K^*$. In parenthesis is the average number of iterations for CG, and the dimension of the approximation space for the Extended Krylov subspace method. For significantly larger problems, we expect the fully iterative version of the Extended Krylov method to be the best performing method.

5. **Transfer function.** The procedure described in the previous sections can be used to approximate the transfer function

$$h(\sigma) = C^*(A - i\sigma E)^{-1}B, \quad \sigma \in [\alpha, \beta].$$

The approximation is performed by determining smaller representation matrices $\tilde{A}$, $\tilde{E}$, $\tilde{B}$ and $\tilde{C}$, such that $h(\sigma) = \tilde{C}^*(\tilde{A} - i\sigma \tilde{E})^{-1}\tilde{B}$ approximates $h(\sigma)$ for all $\sigma$’s in an interval of interest. A crucial feature of any approximation strategy is that, in spite of the size of the original problem, the approximation space should be as small as possible, so as to truly perform a model reduction; see, e.g., [2, 32]. In particular, all relevant oscillatory features of $h(\sigma)$ should be retained in the reduced model.
By building the Extended Krylov subspace associated with $AE^{-1}$ and $B$, one can construct the matrices $V_m, T_m$ and then project the problem, so that
\[
h(\sigma) \approx (V_m^*(E^{-*}C))^*(T_m - i\sigma I)^{-1}E_1\beta_0,
\] with $B = V\beta$.

We stress that we assume here that both $A$ and $E$ are nonsingular. In case of singularity of one of the two matrices, say $A$, we can proceed by translating the problem so that $A - i\sigma_0 E$ is nonsingular, and then work with the transfer function $h(\sigma) = C^*((A - i\sigma_0 E) - i\delta E)^{-1}B$, $\sigma = \sigma_0 + \delta$.

Similar projection procedures have been classically explored in the control literature for approximating the transfer function, leading, e.g. for $E = I$, to the use of the standard Krylov space $K_m(A, B)$, the “shift-and-inverted” Krylov space $K_m((A - \omega I)^{-1}, B)$, and the rational Krylov subspace $\text{span}\{B, (A - \omega_1 I)^{-1}B, (A - \omega_2 I)^{-1}B, \ldots\}$; we refer to [1] for a thorough analysis of these methods and for an exhaustive list of references. By comparing the Extended and rational Krylov spaces, it appears that the Extended Krylov subspace is nothing but a rational space with repeated poles at zero and infinity, for which a theoretical framework may be found in [1, Cor. 11.9, p.355]. On the other hand, it is interesting that apparently an iterative approach to generate this space was never implemented, although it provides some clear advantages over the general rational space, whose pole selection remains in most cases a trial-and-error procedure [1, 15, 16].

![Figure 5.1](image.png)

**Fig. 5.1.** Example 5.1. Bode plot of the first input and output: approximation space dimension equal to 20.

**Example 5.1.** We consider a famous small benchmark problem, the CD Player, available in the SLICOT Library [3], whose matrix $A$ has size $n = 120$ (here $E = I$). The matrices $B$ and $C$ have 2 columns each. For this experiment we have decided to approximate the columns of $B, C$ separately. The block version of the methods could be used to approximate the whole transfer function matrix.

In Figure 5.1 we report the Bode plot, namely the log-log graph of the function $|h_{i,j}(\sigma)|$, associated with the problem for the first input and output. The curves represent the true function (circled thin line), the approximation obtained by the standard Krylov subspace of size 20 (dashed line), by the inverted Krylov subspace of size 20 (dash-dotted line) and by the Extended Krylov subspace of size 20 (solid line).
Clearly, the Extended method is better capable of capturing the oscillations in both parts of the spectrum by combining the features of both the standard and inverted approaches, thus providing an overall optimal approximation of the transfer function. In Figure 5.2 we report the Bode plot associated with the problem for the first input and second output and a subspace of dimension 20 in both cases (left plot). The standard and inverted spaces are unable to approximate most of the curve for such a small dimension, whereas the Extended space does a good job around its poles. However, the overall approximation is still unsatisfactory. The right plot shows the quality of the approximation for a subspace of dimension 50, which allows the Extended Krylov subspace to satisfactorily capture all transfer function’s features. The standard procedure is still missing most of the action in the middle of the frequency interval; the inverted method is still not accurate for high frequencies. We stress here that for the same space dimension, the inverted Krylov space requires twice as many matrix solves as the Extended Krylov space.

Example 5.2. We consider the Benchmark problem Beam [8], which has matrix $A$ of size 348 (also here $E = I$) and single column input-output matrices. Figure 5.3 represents the corresponding transfer function for frequencies in $[10^{-2}, 10^3]$, and its reduced model in a subspace of dimension $m = 20$ (left) and $m = 40$ (right) with both the standard and Extended Krylov spaces, the latter choice delivering a far more accurate approximation of the original function. The inverted Krylov space improves its accuracy on high frequencies when going from $m = 20$ to $m = 40$, but remains unsatisfactory, compared to the combined approach.

6. Conclusions. We have shown that the Extended Krylov subspace may be a competitive alternative to other Krylov-type spaces for solving a variety of algebraic problems of great interest in applications. The effectiveness of the method relies on efficient solves with the involved large-scale nonsingular matrix $A$. We have shown that many advanced scientific problems are so complex that they can afford such a cost. For larger or denser problems, systems with $A$ may be solved by means of preconditioned iterative techniques, giving rise to so-called inner-outer iterations; we refer to [24, 35] for examples of methods based on the Extended Krylov subspace.
The Extended Krylov subspace for parameter dependent systems

Fig. 5.3: Example 5.2 Bode plot of the second input and first output: approximation space dimension equal to 20 (left), and to 40 (right).

Acknowledgments. This work was mostly performed while the author was visiting the TU Berlin during the fall semester 2008. We thank the referees for their constructive criticism which helped improve the presentation of the numerical evidence.

REFERENCES


