A Krylov Subspace Method to Solve a Sequence of Linear Systems with Different Right-Hand Sides

Daniel Skoogh *

November 1998

Abstract

A new method to solve linear systems of equations with several right-hand sides is described. It uses the basis from a previous solution to reduce the number of matrix vector products needed to solve a linear system of equations with a new right-hand side. It builds up a subspace of a union of Krylov spaces. Some numerical examples are given where variants of the method are compared to Krylov subspace methods, particularly a block Arnoldi (GMRES) algorithm.

Keywords: Arnoldi, Krylov, block, right-hand sides, iterative
AMS subject classification 65F10

1 Introduction

Consider a set of k linear systems with the same matrix $\boldsymbol{A}$, but with several right hand sides $\boldsymbol{b}_i$, $i = 1, \dots, k$:

$$\boldsymbol{A}\boldsymbol{x}_i = \boldsymbol{b}_i,\ i = 1, \dots, k \tag{1}$$

$$\boldsymbol{A} \in \mathcal{C}^{n \times n},\ \boldsymbol{b}_i, \boldsymbol{x}_i \in \mathcal{C}^n. \tag{2}$$

We are going to describe a new method that reduces the total number of matrix vector products needed to solve the system (1) compared to when the system is solved for each right-hand side separately. We will discuss cases when all right hand sides are known before the iterations begin, as well as cases when they are not.

If all right-hand sides are known before the iterations begin, then the block Arnoldi method can be used to solve the set of systems (1). It generates orthonormal basis vectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_{m+k}$ that span the subspace

$$\mathcal{S}_{m+k} = \bigcup_{i=1}^{k} \mathcal{K}_{(\frac{m}{k}+1)}(\boldsymbol{A}, \boldsymbol{b}_i), \tag{3}$$

*Report No 1998-46 in "Blå serien" preprint series, available at URL: `http://www.math.chalmers.se/Math/Research/Preprints`. Author's address: Department of Mathematics, Chalmers University of Technology and the University of Göteborg, S-41296 Göteborg, Sweden (`skoogh@math.chalmers.se`) which is a union of Krylov spaces. As usual, a Krylov subspace is defined by

$$\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) = \operatorname{span}\{\boldsymbol{v}, \boldsymbol{A}\boldsymbol{v}, \boldsymbol{A}^2\boldsymbol{v}, \dots, \boldsymbol{A}^{k-1}\boldsymbol{v}\}. \tag{4}$$

In this report we will introduce new algorithms that generate orthonormal basis vectors which span a subspace of a union of Krylov spaces, whereas block Arnoldi generates basis vectors that span the whole union (3). The new algorithms are more flexible, and the convergence behaviour will depend on how the subspace is chosen. We reduce the number of matrix vector products needed to solve a linear system for a new right-hand side by using bases of previous systems.

In section 2 we describe a band and a staircase algorithm where the basis vectors span a union of Krylov subspaces. The band algorithm starts on k orthonormal vectors which are a basis of the subspace spanned by the k right-hand sides, and generates basis vectors that span the same subspace as a block Arnoldi method. The only difference is that the basis is expanded by one vector at a time, whereas in a block Arnoldi method the basis is expanded by k vectors at a time. The staircase algorithm on the other hand starts on the first right-hand side, and the other right-hand sides are incorporated one after another into the basis at later stages of the iterations. Each time a new right-hand $\boldsymbol{b}_i$ is incorporated into the basis, the bandwidth of the computed matrix in the new basis is expanded by one.

In section 3 we make extensions to those two basic Krylov methods where the matrix $\boldsymbol{A}$ operates on a linear combination of the basis vectors. This will cause the basis vectors generated by the algorithms to span a subspace of a union of Krylov spaces.

In section 4 we show test results for these new algorithms and compare them to the basic Krylov methods.

In section 5 we draw conclusions as to when the algorithms developed in this report are suitable for practical use.

1.1 Related Work

The first Krylov subspace method was developed by Lanczos [5], another early work was done by Arnoldi [2]. The first methods built on multiple Krylov subspaces were developed by Cullum and Donath [4] for the eigenvalue problem, and by O'Leary [8] for linear systems. Krylov subspace methods for linear systems are described in detail in the book by Saad [16].

Parlett [10] described how new right-hand sides could be incorporated into a basis. This is further discussed by Saad [15]. Saad calls the method modified Lanczos algorithm. The method is intended for symmetric matrices. Orthogonal projection is used to solve the approximate problem. A new direction is orthogonalised against the basis $\boldsymbol{v}_1, \dots, \boldsymbol{v}_j$ and the normalised result replaces $\boldsymbol{v}_{j+1}$. This method will work well only if the residual has a small part in $\boldsymbol{v}_{j+1}$. Neither Parlett nor Saad have reported any numerical tests in their work.

Schmitt and Weiner describe methods that incorporate new right-hand sides into the basis in [18]. Their methods are intended for unsymmetric matrices. Orthogonal projection is used to solve the approximate problem. For one of the approaches described, the matrix operates on the basis vector that has the largest part of the residual.

Several works [3, 19, 9, 20] deal with the selection of one seed system that is solved by a Krylov method. The other systems are solved by using orthogonal projection onto the subspace generated by the seed system. The procedure then continues either with solving each system individually [9], or by repeating the selection of seed system [3, 19, 20], and using orthogonal projection for the other systems onto the seed system until all systems have converged. The advantage with these methods is that the memory requirement for the basis vectors and the time spent in (bi)orthogonalisation can be kept low by using restarts or by developing recurrences for the solutions. The disadvantage is that these methods are best suited for problems where the right-hand sides are known before the iterations begin.

In our work, we have not assumed that the right-hand sides are close in any way. Some methods may take advantage of right-hand sides that are close, for example in time stepping.

Notation

Matrices are written with upper-case bold letters like $\boldsymbol{A}$, vectors are written with lower-case bold letters like $\boldsymbol{v}$ and scalars are written with lower-case letters like μ. By $h_{i,j}$ we mean the element $\boldsymbol{H}(i,j)$, by $\boldsymbol{H}_{i,j}$ we mean the leading $i \times j$ submatrix of $\boldsymbol{H}$, and by $\boldsymbol{h}_j$ we mean the j:th column of $\boldsymbol{H}$. By $\boldsymbol{V}_j$ we mean the first j columns of the matrix $\boldsymbol{V}$.

2 Basic Krylov Subspace Methods

The algorithms described in this section are variations of the block Arnoldi algorithm. They are given here as reference methods since they build on established theory, although the goal of this report is to extend the theory to the methods we will describe in the next section.

2.1 Band Method

The basis vectors generated by this algorithm span the same subspace (3) as a standard block Arnoldi algorithm, provided that m/k is an integer. In this form the method was developed by Ruhe and is described in [11] for Hermitian matrices. The matrix $\boldsymbol{A}$ operates on one basis vector at a time, whereas in a block method the matrix $\boldsymbol{A}$ operates on several basis vectors at a time.

Band Algorithm

```
1   Choose $k$ orthonormal vectors $\boldsymbol{v}_i,\ i = 1:k$
2   for $j = 1:m$
3       $\boldsymbol{r} = \boldsymbol{A}\boldsymbol{v}_j$ (operate)
4       $[\boldsymbol{v}_{j+k}, \boldsymbol{h}_j] = \textbf{GramSchmidt}(\boldsymbol{V}_{j+k-1}, \boldsymbol{r})$
5       Solve approximate problem
6   end
```

The number of matrix vector products m is chosen dynamically according to some stopping criterion, for example when the norms of all residuals $\boldsymbol{r}_i = \boldsymbol{b}_i - \boldsymbol{A}\tilde{\boldsymbol{x}}_i,\ i = 1, \ldots, k$ are below some threshold value. We will explain how the approximate solution is calculated at line 5 of the algorithm in section 2.3.

The GramSchmidt function is just the standard Gram-Schmidt. However, for the numerical tests to ensure that the basis is orthonormal to working precision, the vector $\boldsymbol{r}$ is orthogonalised twice against the basis.

Gram-Schmidt

```
function $[\boldsymbol{v}, \boldsymbol{h}] =$ GramSchmidt$(\boldsymbol{V}, \boldsymbol{r})$
if $\boldsymbol{V} = []$ then
    $h = \| \boldsymbol{r} \|$
    $\boldsymbol{v} = \boldsymbol{r}/h$
else
    $\boldsymbol{h} = \boldsymbol{V}^H \boldsymbol{r}$ (orthogonalise)
    $\boldsymbol{r} = \boldsymbol{r} - \boldsymbol{V}\boldsymbol{h}$
    $j =$ length$(\boldsymbol{h})$
    $h_{j+1} = \| \boldsymbol{r} \|$
    $\boldsymbol{v} = \boldsymbol{r}/h_{j+1}$ (new basis vector)
end if
return
```

After j iterations with the band algorithm, the Arnoldi factorisation

$$\boldsymbol{A}\boldsymbol{V}_j = \boldsymbol{V}_{j+k}\boldsymbol{H}_{j+k,j} \tag{5}$$

holds. The $(j+k) \times j$ trapezoidal matrix $\boldsymbol{H}$ has lower bandwidth k.

The starting vectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_k$ are an orthonormal basis of the subspace spanned by the right-hand sides $\boldsymbol{b}_1, \dots, \boldsymbol{b}_k$. They are computed by a QR factorisation

$$\boldsymbol{B} = \boldsymbol{V}_k\boldsymbol{S},\ \boldsymbol{B} \in \mathcal{C}^{n\times k},\ \boldsymbol{V}_k \in \mathcal{C}^{n\times k},\ \boldsymbol{S} \in \mathcal{C}^{k\times k}. \tag{6}$$

2.2 Staircase Method

This algorithm starts on one vector, the normalised first right-hand side $\boldsymbol{v}_1 = \boldsymbol{b}_1 / \| \boldsymbol{b}_1 \|$, and the other right-hand sides are incorporated into the basis one after another at later stages of the iteration. This algorithm is described by Schmitt and Weiner [18].

Staircase Algorithm

```
$j = 0$
for $i = 1 : k$
    $[\boldsymbol{v}_{j+i}, \boldsymbol{s}_i] =$ GramSchmidt$(\boldsymbol{V}_{j+i-1}, \boldsymbol{b}_i)$
    for $l = 1 : m_i$
        $j = j + 1$
        $\boldsymbol{r} = \boldsymbol{A}\boldsymbol{v}_j$ (operate)
        $[\boldsymbol{v}_{j+i}, \boldsymbol{h}_j] =$ GramSchmidt$(\boldsymbol{V}_{j+i-1}, \boldsymbol{r})$
        Solve approximate problem
    end
end
```

The number of matrix vector products m_i for each incorporated right-hand side is chosen dynamically, according to some stopping criterion, for example when the norm of the residual $\boldsymbol{r}_i = \boldsymbol{b}_i - \boldsymbol{A}\tilde{\boldsymbol{x}}_i$ is below some threshold value.

After m_1 iterations in the loop on lines 4 to 9, the usual Arnoldi factorisation holds:

$$\boldsymbol{A}\boldsymbol{V}_{m_1} = \boldsymbol{V}_{m_1+1}\boldsymbol{H}_{m_1+1,m_1}.$$

Before the iterations continue, the next right-hand side $\boldsymbol{b}_2$ is orthogonalised against the basis by the Gram-Schmidt and the basis is expanded with the resulting vector $\boldsymbol{v}_{m_1+2}$. A row of zeros is added to the matrix $\boldsymbol{H}_{m_1+1,m_1}$ and the factorisation will be

$$\boldsymbol{A}\boldsymbol{V}_{m_1} = \boldsymbol{V}_{m_1+2}\boldsymbol{H}_{m_1+2,m_1}$$

similar to block Arnoldi with block size 2. The iterations will continue with the expanded matrix $\boldsymbol{V}_{m_1+2}$. Each time a new vector is included in line 3 of the algorithm, the lower bandwidth of the matrix $\boldsymbol{H}$ is increased by one in the iterations that follow. See Figure 9 for a typical structure of the matrix $\boldsymbol{H}$. At step j and the inclusion of i new right-hand sides, the relation

$$\boldsymbol{A}\boldsymbol{V}_j = \boldsymbol{V}_{j+i}\boldsymbol{H}_{j+i,j} \tag{7}$$

holds.

2.3 Approximate Solution

We see from (5) and (7) that the band algorithm and the staircase algorithm have similar Arnoldi factorisations. We can can compute the approximate solutions in the same way for both methods. However, they will have different convergence properties.

From line 3 of the staircase algorithm and from the initial QR factorisation (6) of the band algorithm, we see that the right-hand sides $\boldsymbol{b}_l$, $l = 1, \dots, i$ can be expressed as

$$\boldsymbol{b}_l = \boldsymbol{V}_{j+i}\boldsymbol{s}_l \tag{8}$$

with the appropriate number of zeros added to the bottom of $\boldsymbol{s}_l$, $l = 1, \dots, i$. If for each $l = 1, \dots, i$ we solve the problem

$$\min_{\boldsymbol{y}_l} \| \boldsymbol{s}_l - \boldsymbol{H}_{j+i,j}\boldsymbol{y}_l \|_2$$

and take

$$\tilde{\boldsymbol{x}}_l = \boldsymbol{V}_{j+i}\boldsymbol{y}_l \tag{9}$$

as the approximate solution, then the residual will be

$$\begin{aligned} \boldsymbol{r}_l &= \boldsymbol{b}_l - \boldsymbol{A}\tilde{\boldsymbol{x}}_l \\ &= \boldsymbol{V}_{j+i}\boldsymbol{s}_l - \boldsymbol{A}\boldsymbol{V}_j\boldsymbol{y}_l \\ &= \boldsymbol{V}_{j+i}\boldsymbol{s}_l - \boldsymbol{V}_{j+i}\boldsymbol{H}_{j+i,j}\boldsymbol{y}_l \\ &= \boldsymbol{V}_{j+i}(\boldsymbol{s}_l - \boldsymbol{H}_{j+i,j}\boldsymbol{y}_l). \end{aligned}$$

The second equality follows from (8) and (9), and the third from (7).

This corresponds to the standard GMRES algorithm by Saad and Schultz [17].

2.4 Subspaces

Both the band algorithm and the staircase algorithm build up bases that span a union of Krylov spaces.

The basis vectors of the band algorithm span the subspace

$$\mathcal{S}_{m+k} = \bigcup_{i=1}^{k} \mathcal{K}_{(\frac{m}{k}+1)}(\boldsymbol{A}, \boldsymbol{b}_i)$$

where m is the total number of matrix vector products and $\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v})$ stands for a Krylov subspace (4).

The basis vectors of the staircase algorithm span the subspace

$$\mathcal{S}_{m+k} = \bigcup_{i=1}^{k} \mathcal{K}_{q_i}(\boldsymbol{A}, \boldsymbol{b}_i),\ q_i = 1 + \sum_{l=i}^{k} \frac{m_l}{l},$$

where now m, the total number of matrix vector products, is

$$m = \sum_{i=1}^{k} m_i.$$

3 Adaptive Methods

We will now extend the two basic Krylov algorithms to operate on linear combinations of the basis vectors. This is similar to the rational Krylov algorithm developed by Ruhe [12, 13, 14], for eigenvalue computation.

Adaptive Band Algorithm

```
Choose $k$ orthonormal vectors $\boldsymbol{v}_i,\ i = 1 : k$
**for** $j = 1 : m$
    Choose continuation combination $\boldsymbol{V}_{j+k-1}\boldsymbol{t}_j$
    $\boldsymbol{r} = \boldsymbol{A}\boldsymbol{V}_{j+k-1}\boldsymbol{t}_j$ (operate)
    $[\boldsymbol{v}_{j+k}, \boldsymbol{h}_j] = \textbf{GramSchmidt}(\boldsymbol{V}_{j+k-1}, \boldsymbol{r})$
    Solve approximate problem
**end**
```

In the algorithm above, the basis vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$ are an orthonormal basis of the subspace spanned by the right-hand sides, just as in the band algorithm.

Adaptive Staircase Algorithm

```
$j = 0$
for $i = 1 : k$
    $[\boldsymbol{v}_{j+i}, \boldsymbol{s}_i] =$ GramSchmidt$(\boldsymbol{V}_{j+i-1}, \boldsymbol{b}_i)$
    for $l = 1 : m_i$
        $j = j + 1$
        Choose continuation combination $\boldsymbol{V}_{j+i-1}\boldsymbol{t}_j$
        $\boldsymbol{r} = \boldsymbol{A}\boldsymbol{V}_{j+i-1}\boldsymbol{t}_j$ (operate)
        $[\boldsymbol{v}_{j+i}, \boldsymbol{h}_j] =$ GramSchmidt$(\boldsymbol{V}_{j+i-1}, \boldsymbol{r})$
        Solve approximate problem
    end
end
```

For both algorithms the choice of continuation combination $\boldsymbol{V}_{j+i-1}\boldsymbol{t}_j$ is essential for the success of the algorithms; it is discussed in section 3.3.

Let us now see how the original problem is expressed in the new basis $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{j+i}$, where j is the total number of matrix vector products, and i is the number of incorporated right-hand sides. From the algorithms above, we see that in step j

$$\boldsymbol{r} = \boldsymbol{A}\boldsymbol{V}_{j+i-1}\boldsymbol{t}_j$$

and Gram-Schmidt implies that

$$\boldsymbol{r} = \boldsymbol{V}_{j+i}\boldsymbol{h}_j$$

where $\boldsymbol{h}_j$ is a vector of length $j+1$. Add a zero to the bottom of $\boldsymbol{t}_j$ and denote

$$\boldsymbol{t}_j = \begin{bmatrix} \boldsymbol{t}_j \\ 0 \end{bmatrix},$$

and we get

$$\boldsymbol{A}\boldsymbol{V}_{j+i}\boldsymbol{t}_j = \boldsymbol{V}_{j+i}\boldsymbol{h}_j.$$

Join the columns and get

$$\boldsymbol{A}\boldsymbol{V}_{j+i}\boldsymbol{T}_{j+i,j} = \boldsymbol{V}_{j+i}\boldsymbol{H}_{j+i,j}.$$

Note that $\boldsymbol{T}$ is not necessarily triangular, but the index of the last nonzero in each column of the matrix $\boldsymbol{T}$ is one less than the corresponding index of the matrix $\boldsymbol{H}$; compare Figures 9 and 10 for the adaptive staircase algorithm.

If the matrix $\boldsymbol{T}$ is chosen to be the unit matrix, the algorithms discussed in this section are identical to those discussed in the previous section.

3.1 Approximate Solution

Let us now describe how to solve the approximate problem at step j. Assume that we have incorporated i right-hand sides. Factorising the matrix $\boldsymbol{T}_{j+i,j}$ with a QR factorisation

$$\boldsymbol{T}_{j+i,j} = \boldsymbol{Q}_{j+i,j}\boldsymbol{R}_{j,j}$$

and setting

$$\hat{\boldsymbol{H}}_{j+i,j} = \boldsymbol{H}_{j+i,j}\boldsymbol{R}_{j,j}^{-1}$$

the basic recursion (3) then becomes

$$\boldsymbol{A}\boldsymbol{V}_{j+i}\boldsymbol{Q}_{j+i,j} = \boldsymbol{V}_{j+i}\hat{\boldsymbol{H}}_{j+i,j}. \tag{10}$$

Just as in the previous section, the right-hand sides $\boldsymbol{b}_l,\ l = 1, \ldots i$, can be expressed as

$$\boldsymbol{b}_l = \boldsymbol{V}_{j+i}\boldsymbol{s}_l \tag{11}$$

with the appropriate number of zeros added at the bottom of $\boldsymbol{s}_l,\ l = 1, \ldots, i$. If for each $l = 1, \ldots, i$ we solve the problem

$$\min_{\boldsymbol{y}_l} \parallel \boldsymbol{s}_l - \hat{\boldsymbol{H}}_{j+i,j}\boldsymbol{y}_l \parallel_2 \tag{12}$$

and take

$$\tilde{\boldsymbol{x}}_l = \boldsymbol{V}_{j+i}\boldsymbol{Q}_{j+i,j}\boldsymbol{y}_l \tag{13}$$

as the approximate solution, then the residual will be

$$\begin{aligned} \boldsymbol{r}_l &= \boldsymbol{b}_l - \boldsymbol{A}\tilde{\boldsymbol{x}}_l \\ &= \boldsymbol{V}_{j+i}\boldsymbol{s}_l - \boldsymbol{A}\boldsymbol{V}_{j+i}\boldsymbol{Q}_{j+i,j}\boldsymbol{y}_l \\ &= \boldsymbol{V}_{j+i}\boldsymbol{s}_l - \boldsymbol{V}_{j+i}\hat{\boldsymbol{H}}_{j+i,j}\boldsymbol{y}_l \\ &= \boldsymbol{V}_{j+i}(\boldsymbol{s}_l - \hat{\boldsymbol{H}}_{j+i,j}\boldsymbol{y}_l). \end{aligned} \tag{14}$$

The second equality follows from (11) and (13), and the third from (10). Since $\boldsymbol{V}_{j+i}$ has orthonormal columns, we have for each $l = 1, \ldots, i$

$$\parallel \boldsymbol{b}_l - \boldsymbol{A}\tilde{\boldsymbol{x}}_l \parallel_2 = \parallel \boldsymbol{s}_l - \hat{\boldsymbol{H}}_{j+i,j}\boldsymbol{y}_l \parallel_2 .$$

3.1.1 Solution by Orthogonal Projection

Expand the matrix $\boldsymbol{Q}_{j+i,j}$ with i columns into an orthogonal matrix $\boldsymbol{Q}_{j+i,j+i}$. Then (10) becomes

$$\boldsymbol{A}\boldsymbol{V}_{j+i}\boldsymbol{Q}_{j+i,j} = \boldsymbol{V}_{j+i}\boldsymbol{Q}_{j+i,j+i}\boldsymbol{Q}_{j+i,j+i}^H\hat{\boldsymbol{H}}_{j+i,j}.$$

Set

$$\boldsymbol{W}_{j+i} = \boldsymbol{V}_{j+i}\boldsymbol{Q}_{j+i,j+i} \tag{15}$$

and

$$\begin{aligned} \hat{\hat{\boldsymbol{H}}}_{j+i,j} &= \boldsymbol{Q}_{j+i,j+i}^H\hat{\boldsymbol{H}}_{j+i,j} \\ &= \boldsymbol{Q}_{j+i,j+i}^H\boldsymbol{H}_{j+i,j}\boldsymbol{R}_{j,j}^{-1} \end{aligned} \tag{16}$$

whence we get

$$\boldsymbol{A}\boldsymbol{W}_j = \boldsymbol{W}_{j+i}\hat{\hat{\boldsymbol{H}}}_{j+i,j}. \tag{17}$$

If the matrix $\boldsymbol{A}$ is Hermitian, then the matrix $\hat{\hat{\boldsymbol{H}}}_{j,j}$ is Hermitian; further, from (16) we see that the matrix $\boldsymbol{H}_{j,j}$ is not Hermitian in general and the vector $\boldsymbol{r} = \boldsymbol{A}\boldsymbol{V}_{j+i-1}\boldsymbol{t}_j$ needs to be orthogonalised against all basis vectors.

For the right-hand sides $\boldsymbol{b}_l, l = 1, \dots, i$ we find that even if

$$\boldsymbol{b}_l \in \operatorname{span}\{\boldsymbol{v}_1, \dots, \boldsymbol{v}_j\},$$

the right-hand side $\boldsymbol{b}_l$ does not belong to the subspace

$$\operatorname{span}\{\boldsymbol{w}_1, \dots, \boldsymbol{w}_j\}$$

in general, because

$$\boldsymbol{W}_j = \boldsymbol{V}_{j+i}\boldsymbol{Q}_{j+i,j}.$$

This means that important information is kept in all the basis vectors $\boldsymbol{w}_1, \dots, \boldsymbol{w}_{j+i}$. If we solve the approximate problem by an orthogonal projection method

$$\begin{aligned}
\boldsymbol{W}_j^H(\boldsymbol{b}_l - \boldsymbol{A}\tilde{\boldsymbol{x}}_l) &= \boldsymbol{W}_j^H(\boldsymbol{b}_l - \boldsymbol{A}\boldsymbol{W}_j\boldsymbol{y}_l) \\
&= \boldsymbol{W}_j^H\boldsymbol{b}_l - \boldsymbol{W}_j^H\boldsymbol{A}\boldsymbol{W}_j\boldsymbol{y}_l \\
&= \boldsymbol{W}_j^H\boldsymbol{b}_l - \hat{\hat{\boldsymbol{H}}}_{j,j}\boldsymbol{y}_l \\
&= 0,
\end{aligned}$$

then the approximate solution $\tilde{\boldsymbol{x}}_l = \boldsymbol{W}_j\boldsymbol{y}_l$ may in some cases not be a good approximation, because the last i basis vectors $\boldsymbol{w}_{j+1}, \dots, \boldsymbol{w}_{j+i}$ will not be used to compute the approximate solution.

3.1.2 Solution by Residual Minimisation

An alternative way of calculating the approximate solution (13) can be derived from the modified Arnoldi factorisation (17).

If for each $l = 1, \dots, i$ we set $\hat{\boldsymbol{s}}_l = \boldsymbol{W}_{j+i}^H\boldsymbol{b}_l$, and solve the problem

$$\min_{\boldsymbol{y}_l} \| \hat{\boldsymbol{s}}_l - \hat{\hat{\boldsymbol{H}}}_{j+i,j}\boldsymbol{y}_l \|_2,$$

and take $\tilde{\boldsymbol{x}}_l = \boldsymbol{W}_j\boldsymbol{y}_l$ as the approximate solution, then the residual will be

$$\begin{aligned}
\boldsymbol{b}_l - \boldsymbol{A}\tilde{\boldsymbol{x}}_l &= \boldsymbol{W}_{j+i}\hat{\boldsymbol{s}}_l - \boldsymbol{A}\boldsymbol{W}_j\boldsymbol{y}_l \\
&= \boldsymbol{W}_{j+i}\hat{\boldsymbol{s}}_l - \boldsymbol{W}_{j+i}\hat{\hat{\boldsymbol{H}}}_{j+i,j}\boldsymbol{y}_l \qquad (18) \\
&= \boldsymbol{W}_{j+i}(\hat{\boldsymbol{s}}_l - \hat{\hat{\boldsymbol{H}}}_{j+i,j}\boldsymbol{y}_l)
\end{aligned}$$

where the second equality comes from (17). Note that

$$\begin{aligned}
\boldsymbol{b}_l - \boldsymbol{A}\tilde{\boldsymbol{x}}_l &= \boldsymbol{V}_{j+i}\boldsymbol{Q}_{j+i,j+i}(\hat{\boldsymbol{s}}_l - \hat{\hat{\boldsymbol{H}}}_{j+i,j}\boldsymbol{y}_l) \\
&= \boldsymbol{V}_{j+i}(\boldsymbol{Q}_{j+i,j+i}\hat{\boldsymbol{s}}_l - \boldsymbol{Q}_{j+i,j+i}\hat{\hat{\boldsymbol{H}}}_{j+i,j}\boldsymbol{y}_l) \\
&= \boldsymbol{V}_{j+i}(\boldsymbol{Q}_{j+i,j+i}\boldsymbol{Q}_{j+i,j+i}^H\boldsymbol{V}_{j+i}^H\boldsymbol{b}_l - \boldsymbol{Q}_{j+i,j+i}\boldsymbol{Q}_{j+i,j+i}^H\hat{\boldsymbol{H}}_{j+i,j}\boldsymbol{y}_l) \\
&= \boldsymbol{V}_{j+i}(\boldsymbol{s}_l - \hat{\boldsymbol{H}}_{j+i,j}\boldsymbol{y}_l)
\end{aligned}$$

which is equivalent to (14). The first equality follows from (15) and (18), and the third from that $\hat{\boldsymbol{s}}_l = \boldsymbol{W}_{j+i}^H \boldsymbol{b}_l$ and the definition (16) of $\hat{\hat{\boldsymbol{H}}}_{j+i,j}$.

3.2 Subspaces

In both the adaptive band algorithm and the adaptive staircase algorithm, the matrix $\boldsymbol{A}$ operates on a linear combination $\boldsymbol{Vt}$ of the basis vectors $\boldsymbol{V}$. As a result, for each of the two algorithms, the basis vectors that are generated span a subspace of a union of Krylov spaces. Note that the band algorithm and the staircase algorithm discussed in section 2 generate basis vectors that span a full union of Krylov spaces.

The basis vectors of the adaptive band algorithm span a subspace of dimension $m + k$ of the space

$$\mathcal{S}_{km+k} = \bigcup_{i=1}^{k} \mathcal{K}_{m+1}(\boldsymbol{A}, \boldsymbol{b}_i)$$

where m is the total number of matrix vector products. The basis vectors of the adaptive staircase algorithm span a subspace of dimension $m+k$ of the subspace

$$\mathcal{S}_f = \bigcup_{i=1}^{k} \mathcal{K}_{q_i}(\boldsymbol{A}, \boldsymbol{b}_i),\ f = \sum_{l=1}^{k} l m_l + k,\ q_i = 1 + \sum_{l=i}^{k} m_l$$

where now m, the total number of matrix vector products, is

$$m = \sum_{i=1}^{k} m_i.$$

3.3 Continuation Vector

The continuation vectors $\boldsymbol{t}_j$ are essential for the success of the algorithms. For the first right-hand side ($i = 1$) and $\boldsymbol{t}_{j,j} \neq 0$, the basis vectors span a Krylov subspace, and choice of continuation vectors is not critical. For later right-hand sides ($i > 1$), different choices of continuation vectors $\boldsymbol{t}_j$ will generate different subspaces.

We have had success continuation on a normalised residual vector. Solve the small problem (12) and take $\boldsymbol{t}_j$ as

$$\boldsymbol{t}_j = \frac{\boldsymbol{s}_l - \hat{\boldsymbol{H}}_{j+i,j}\boldsymbol{y}_l}{\| \boldsymbol{s}_l - \hat{\boldsymbol{H}}_{j+i,j}\boldsymbol{y}_l \|_2}.$$

This is closely related to Arnoldi's method for linear systems with one right-hand side. The residual for the FOM (Full Orthogonalisation Method) variant of Arnoldi's method will be

$$\begin{aligned} \boldsymbol{r} &= \boldsymbol{b} - \boldsymbol{A}\tilde{\boldsymbol{x}} \\ &= \beta \boldsymbol{v}_{j+1} \end{aligned}$$

if the right hand-side $\boldsymbol{b}$ belongs to the span of the basis vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_j$. The vector $\boldsymbol{v}_{j+1}$ is the next basis vector and the matrix $\boldsymbol{A}$ will operate on it in the next iteration.

The choice of right-hand side $\boldsymbol{s}_l$ in (12) has a great impact on the convergence behaviour. In the adaptive band algorithm, one possibility is to alternate between the different right-hand sides $\boldsymbol{s}_l, l = 1, \ldots, k$. In the adaptive staircase algorithm, a natural choice is to take the latest incorporated right-hand side in the new basis $\boldsymbol{s}_i$ as the right hand side in the minimisation (12).

See Figure 10 for a typical structure of the $\boldsymbol{T}$ matrix of the adaptive staircase algorithm.

4 Test Results

The algorithms are implemented in MATLAB for the non-normal test and the clustered normal test. For the CEM test, the matrix $\boldsymbol{A}$ is generated by the Fortran program NEC2 [7, 6], and the algorithms are implemented in MATLAB.

4.1 Non-Normal Test

This test matrix is upper triangular with bandwidth 5, and the eigenvalues $a_{i,i} = (i+1)^{\frac{1}{q}} - 1$ along the diagonal. The parameter q controls the compactness of the spectrum; increasing q gives a cluster nearer to zero. The spectrum is similar to a spectrum of a typical preconditioned matrix.

The 5 diagonals above are filled with normally distributed random numbers with mean $-p$ and variance $2p$. The parameter p controls how normal the matrix is; increasing p gives a more non-normal matrix.

In Figure 1 we plot the spectrum of $\boldsymbol{A}$ of dimension $n = 50$ and for the parameter $q = 3$.

In Figures 3, 4, 5 and 6 we show test results for the algorithms, band algorithm, staircase algorithm, adaptive band algorithm and adaptive staircase algorithm respectively.

In these tests we have chosen the dimension of the matrix $\boldsymbol{A}$ to be $n = 2500$ and the parameter values to be $p = 0.2$ and $q = 3$. The objective is to solve the matrix equation for 3 right-hand sides to an accuracy measured by the norm of the residual, to be less than 10^{-10}. All right-hand sides are generated by normally distributed random numbers and are normalised for unit length. This implies that the different right-hand sides are close to being orthonormal. This is the worst case and the methods cannot take advantage of right-hand sides that are close to each other.

In the figures, the norm of the residual is plotted against the number of iterations for the different right-hand sides.

In Figures 3 and 5 we show the test results for the band algorithm and the adaptive band algorithm respectively. The convergence behaviour is similar; the band algorithm converges in 162 iterations and the adaptive band algorithm converges in 157 iterations. Note that the use of the $\boldsymbol{T}$ matrix in the adaptive band algorithm makes it possible to drop a linear system as soon as the norm of its residual is below 10^{-10}, whereas in the band algorithm we must iterate on every linear system until all of them are below 10^{-10}.

In Figure 4, the test result for the staircase algorithm is plotted. First, the method starts on the first right-hand side and iterates until the norm of the residual is below 10^{-10} (74 iterations). Before the iterations proceed, the second right-hand side is incorporated into the basis. After that, the algorithm continues with the iterations until the norm of the residual for the second right-hand side is below 10^{-10} (additional 84 iterations). Note that the residual norm for the linear system number 1 continues to drop until it reaches roundoff. Last, the third right-hand side is incorporated, and the iterations continue until its residual norm is below 10^{-10} (additional 99 iterations). The algorithm needed a total of 257 iterations. Note that the number of iterations to solve a linear system increases with the number of incorporated right-hand sides. It is evident that this algorithm is not better than any of the others, so we leave it out of the further tests.

In Figure 6, the test result for the adaptive staircase algorithm is plotted. The first linear system needs 74 iterations to converge, the second needs 49 iterations and the third needs 37 iterations. The total number of iterations for the method is 160. Note that the residual norms of the first two linear systems never go much below 10^{-10}, and that the later systems need fewer iterations.

In Table 1 we give test results for the band algorithm, the adaptive band algorithm and the adaptive staircase algorithm. The objective in these tests is to solve the matrix equation with six right-hand sides and the residual norms to be smaller than 10^{-10}. The dimension of the matrix $\boldsymbol{A}$ is $n = 2500$ and the parameters p and q are given different values. The total number of iterations is almost the same for the adaptive staircase algorithm and adaptive band algorithm, whereas the band algorithm needs about 5% more iterations. If the matrix gets less normal (the parameter p increases), or the spectrum gets less compact (the parameter q decreases), then the number of iterations increases for all methods and the ratio of the number of iterations for the last right-hand side to the number of iterations for the first right-hand side decreases for the adaptive staircase algorithm.

4.2 Clustered Normal Test

This test matrix is diagonal, and thus the eigenvalue distribution is given by the diagonal elements. The first $n_1 + 1$ diagonal elements (eigenvalues) are placed on a circle in the complex plane with centre 0 and radius r, with $n_1 - 1$ single eigenvalues and one double in $(r, 0)$. The remaining diagonal elements (eigenvalues) are placed on the real line and are chosen similar to the non-normal test.

$$a_{k,l} = \begin{cases} re^{\frac{2\pi(k-1)}{n_1}j} & k = l,\ 1 \le k \le n_1 + 1 \\ (k - (n_1 + 1))^{\frac{1}{q}} & k = l,\ n_1 + 2 \le k \le n \\ 0 & k \ne l, \end{cases}$$

where $j = \sqrt{-1}$. In Figure 2 we show the spectrum for the case where the dimension of the matrix $\boldsymbol{A}$ is $n = 50$, the radius of the circle is $r = 0.8$ and the number of different eigenvalues on the radius of the circle is $n_1 = 10$.

In Figure 7 the test results for the adaptive staircase algorithm are given. In this test, the dimension of the matrix $\boldsymbol{A}$ is $n = 2500$ and the radius of the circle is $r = 0.1$, the number of different eigenvalues on the circle is $n_1 = 10$ and the parameter value is $q = 3$. The algorithm needs 93 iterations to solve the first linear system, 32 to solve the second and 25 to solve the third. The large difference between the number of iterations needed to solve the different linear systems can be explained as follows. All eigenvalues on the disc have modulus $r = 0.1$; the smallest eigenvalue that does not belong to the disc has modulus 1. Thus the eigenvalues on the disc have a great impact on the solution. When the first linear system is solved, the eigenvalues on the disc need to converge with corresponding eigenvectors. When the second linear system is solved, the information about the n_1 eigenvalues with the corresponding eigenvectors is already in the basis; however, there is one double eigenvalue on the disc $(r, 0)$ and it needs to converge with the corresponding eigenvector. When the third linear system is solved, all eigenvalues with corresponding eigenvectors on the disc have already converged, so the third linear system needs even fewer iterations to converge.

In Table 2 we give test results for the band algorithm, the adaptive band algorithm and the adaptive staircase algorithm. The objective in these tests is to solve the matrix equation with six right-hand sides and the residual norms to be less than 10^{-10}. The dimension of the matrix $\boldsymbol{A}$ is $n = 2500$, the parameter value is $q = 3$ and the parameters r (the radius of the disc) and n_1 (the number of different eigenvalues) are given different values. The total number of iterations is almost the same for the adaptive staircase algorithm and adaptive band algorithm, whereas the band algorithm needs about 5% more iterations. If more eigenvalues are placed on the disc (the parameter n_1 increases), or the radius of the disc decreases (the parameter r decreases), then the number of iterations increases for all algorithms and the ratio of the number of iterations for the last right-hand side to the number of iterations for the first right-hand side decreases for the adaptive staircase algorithm.

4.3 CEM Test

This test matrix comes from a computational electromagnetic field problem. It was generated by the numerical electromagnetic code NEC2, developed for the US Navy by Lawrence Livermore Labs in 1981; see [7] and [6]. The analysis in NEC2 is done by numerical solution of integral equations for induced currents. This yields dense complex matrices (double complex in Fortran). The test problem consists of a simple wire model of an airplane; see Figure 8. The wire model is excited with plane time-harmonic electromagnetic waves. The structure of the wire model at a given frequency yield the test matrix. The angle, the polarisation and the frequency of the incident electromagnetic waves yields the different right-hand sides.

In this case we use a sparse preconditioner. A sparse matrix $\hat{\boldsymbol{A}}$ is created from $\boldsymbol{A}$ by

$$\hat{a}_{i,j} = \begin{cases} a_{i,j} & \text{if } |a_{i,j}| > \epsilon_1 \max_{1 \le i,j \le n} |a_{i,j}| \\ 0 & \text{otherwise} \end{cases}$$

The matrix $\boldsymbol{A}$ and thus $\hat{\boldsymbol{A}}$ are nearly symmetric in that $|a_{i,j}| \approx |a_{j,i}|$. The matrix $\hat{\boldsymbol{A}}$ is first ordered by minimum degree (symmmd in MATLAB) and later factorised by incomplete factorisation with drop tolerance ϵ_2 (luinc in MATLAB),

$$\hat{\boldsymbol{L}}\hat{\boldsymbol{U}} \approx \boldsymbol{P}\hat{A}\boldsymbol{P}^t,$$

where $\boldsymbol{P}$ is the permutation matrix. The adaptive staircase algorithm is now applied to the linear system

$$\boldsymbol{P}^t\hat{\boldsymbol{U}}^{-1}\hat{\boldsymbol{L}}^{-1}\boldsymbol{P}\boldsymbol{A}\tilde{\boldsymbol{x}}_i = \boldsymbol{P}^t\hat{\boldsymbol{U}}^{-1}\hat{\boldsymbol{L}}^{-1}\boldsymbol{P}\boldsymbol{b}_i.$$

The wire model of the airplane is excited with an electromagnetic wave of a frequency of 30 MHz. The wires are approximately 0.07 wavelengths apart. This gives a test matrix of dimension $n = 3549$. The different incident electromagnetic waves are orthogonal to each other.

The test problem is solved for three different preconditioners (different values of the parameters ϵ_1 and ϵ_2). The adaptive staircase algorithm is compared to the block Arnoldi algorithm. Block Arnoldi is mathematically equivalent to our band algorithm, but block Arnoldi is faster since it operates on several vectors at a time. The approximate solution is calculated as described in section 2.3.

In Table 3 we show the test results with regard to the number of matrix vector products. The number r_1 is defined as the ratio of filled elements in $\hat{\boldsymbol{A}}$ and r_2 in $\hat{\boldsymbol{L}} + \hat{\boldsymbol{U}}$. From the table we see that the total number of matrix vector products decreases as the preconditioner gets denser (r_1 and r_2 increases). We see also that the ratio m_4/m_1 increases as the preconditioner gets denser. In each test the adaptive staircase algorithm needs approximately 12% fewer matrix vector products than the block Arnoldi algorithm.

In Table 4 we show timing results. The algorithms are implemented in MATLAB and run on one processor on a SUN Enterprise 10000. It is a multiprocessor machine with shared memory. The dominant part is the matrix vector product $\boldsymbol{Ax}$. In each test block Arnoldi needs less than half the time.

In Table 5 we show comparisons between MATLAB and LAPACK/BLAS [1] for matrix vector product and Gaussian elimination. For the matrix vector product, the BLAS routine zgemv is 14 times faster than MATLAB, and the Gaussian elimination routine in LAPACK zgesv is 1.5 times faster than MATLAB. Note that the Gaussian elimination in MATLAB is faster than solving the linear system by the adaptive staircase algorithm.

The tests with regard to timing are somewhat artificial. To get a realistic test one needs to implement the algorithms on a parallel computer with FORTRAN and MPI (message passing interface) and run on a large problem.

5 Conclusions and Extensions

From the test problems we see that both the adaptive band algorithm and the adaptive staircase algorithm perform better than basic Krylov methods in terms of the number of matrix vector products needed to solve the linear systems.

Even though the timing tests of the CEM test are somewhat artificial, it indicates when the adaptive band method and adaptive staircase algorithm can be useful. The methods use all basis vectors in every step, both for creating the continuation combination $\boldsymbol{V}_j\boldsymbol{t}_j$ and for the orthogonalisation. This means that the methods are suitable for problems where the time spent computing the matrix vector product is the dominant part of the algorithm, for example matrices created from integral equations. Further, if the right-hand sides are not known when starting the iterations, the adaptive staircase algorithm can be a suitable choice, since then block Arnoldi cannot be used.

We have not assumed in our work that the right-hand sides are close. If the right-hand sides are close as in time stepping, then the performance of the adaptive staircase algorithm is likely to improve considerably.

When all right-hand sides are known in advance, a combination of the adaptive staircase algorithm and a block method could be used. First the adaptive staircase algorithm is used to solve for the first few right-hand sides. After the initial basis has been built, a block method could be used to solve for the rest of the right-hand sides, all at once or a few of them at a time by deleting the last basis vectors generated by the preceding block run.

Non-Normal Test											
		Adaptive Staircase Algorithm								Band Algorithm	
											Adaptive
p	q	m_1	m_2	m_3	m_4	m_5	m_6	Tot	$\frac{m_6}{m_1}$	m	m
0	2	79	51	42	40	36	35	283	0.44	298	282
0.1	2	84	53	46	38	37	34	292	0.40	306	292
0.2	2	92	53	45	40	37	36	303	0.39	323	303
0.4	2	105	58	46	44	39	36	328	0.34	339	326
0	3	53	36	31	29	28	26	203	0.49	212	202
0.1	3	64	40	35	31	29	28	227	0.44	236	225
0.2	3	74	49	37	34	31	30	255	0.41	263	254
0.4	3	104	57	50	40	39	35	325	0.34	359	320

Table 1: Test results for the band algorithm, adaptive band algorithm and adaptive staircase algorithm applied to the non-normal test solved for six right-hand sides, to residual accuracy 10^{-10}. The dimension of the matrix $\boldsymbol{A}$ is $n = 2500$; the parameters p and q are given different values.

Clustered Normal Test											
		Adaptive Staircase Algorithm								Band Algorithm	
											Adaptive
r	n_1	m_1	m_2	m_3	m_4	m_5	m_6	Tot	$\frac{m_6}{m_1}$	m	m
0.1	0	36	30	27	25	24	23	165	0.64	169	165
0.1	5	65	33	25	24	23	23	193	0.35	212	193
0.1	10	93	32	25	23	23	22	218	0.24	228	217
0.1	20	147	30	23	23	22	21	266	0.14	277	265
0.2	5	61	33	26	24	24	23	191	0.38	203	189
0.2	10	85	32	25	24	23	22	211	0.26	218	211
0.2	20	132	30	24	23	22	22	253	0.17	261	251

Table 2: Test results for the band algorithm, adaptive band algorithm and adaptive staircase algorithm applied to the clustered normal test solved for six right-hand sides, to residual accuracy 10^{-10}. The dimension of the matrix $\boldsymbol{A}$ is $n = 2500$, the parameter $q = 3$, and the parameters n_1 and r are given different values.

CEM Test										
				Adaptive Staircase Algorithm						B. A.
ϵ_1	ϵ_2	r_1	r_2	m_1	m_2	m_3	m_4	Tot	$\frac{m_4}{m_1}$	m
$2.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-3}$	0.034	0.044	190	25	21	17	253	0.09	288
$1 \cdot 10^{-4}$	$1 \cdot 10^{-3}$	0.082	0.093	101	22	18	14	155	0.14	180
$5 \cdot 10^{-5}$	$8 \cdot 10^{-4}$	0.16	0.15	67	19	16	13	115	0.19	136

Table 3: Test results for adaptive staircase algorithm and block Arnoldi algorithm applied to the CEM test, solved for four right-hand sides, to residual accuracy 10^{-7}. The matrix $\boldsymbol{A}$ is a dense complex matrix of dimension $n = 3549$. In the table B. A. stands for Block Arnoldi.

CEM test, timing results (seconds)						
Adaptive Staircase Algorithm						
ϵ_1	$2.5 \cdot 10^{-4}$		$1 \cdot 10^{-4}$		$5 \cdot 10^{-5}$	
ϵ_2	$2.5 \cdot 10^{-3}$		$1 \cdot 10^{-3}$		$8 \cdot 10^{-4}$	
	time	% of tot	time	% of tot	time	% of tot
Ordering	37.1	0.33%	235	3.21%	2250	29.6%
incomplete factorisation	36.4	0.32%	155	2.12%	327	4.31%
matrix vector product $\boldsymbol{Ax}$	9880	87.8%	6250	85.6%	4490	59.1%
apply preconditioner	190	1.7%	260	3.56%	312	4.12%
create start vector ($\boldsymbol{Vt}$)	165	1.46%	59.9	0.82%	32.12	0.42%
Gram-Schmidt	802	7.12%	290	3.96%	155	2.04%
other	146	1.3%	53.4	0.773%	31.2	0.41%
total	12600	100%	7300	100%	7590	100%
Block Arnoldi						
total	4200		3000		4550	

Table 4: Test results for adaptive staircase algorithm and block Arnoldi algorithm applied to the CEM test, solved for four right-hand sides. The time is measured in seconds.

comparing MATLAB with BLAS and LAPACK			
	MATLAB (seconds)	LAPACK/BLAS (seconds)	$\frac{\text{time in MATLAB}}{\text{time in BLAS/LAPACK}}$
matrix vector product ($\boldsymbol{Ax}$)	39.0	2.79 (BLAS)	14.2
Gaussian elimination	4960	3170 (LAPACK)	1.55

Table 5: Comparing MATLAB with BLAS and LAPACK for the CEM test. The tests are done on one processor on a SUN Enterprise 10000. It is a multiprocessor machine with shared memory.

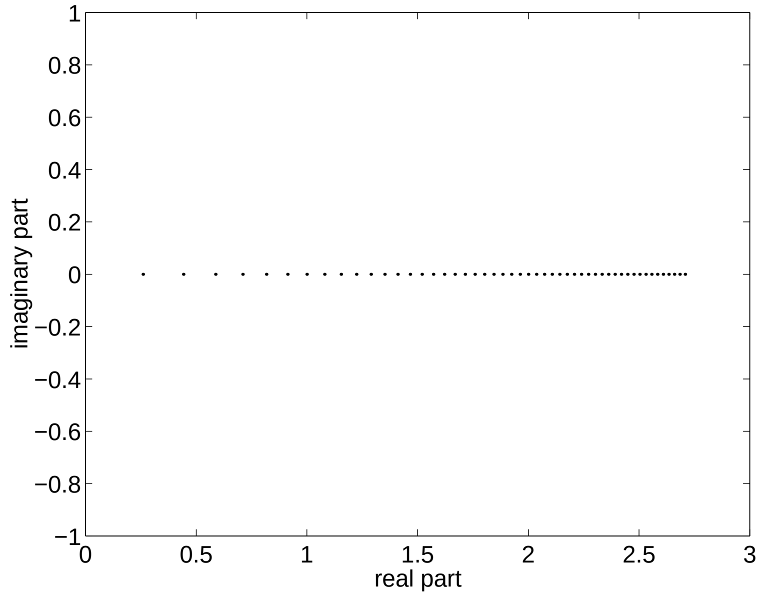

Figure 1: The spectrum of the matrix $\boldsymbol{A}$ for the non-normal test for the case where the dimension of the matrix $\boldsymbol{A}$ is $n = 50$ and the parameter value is $q = 3$.

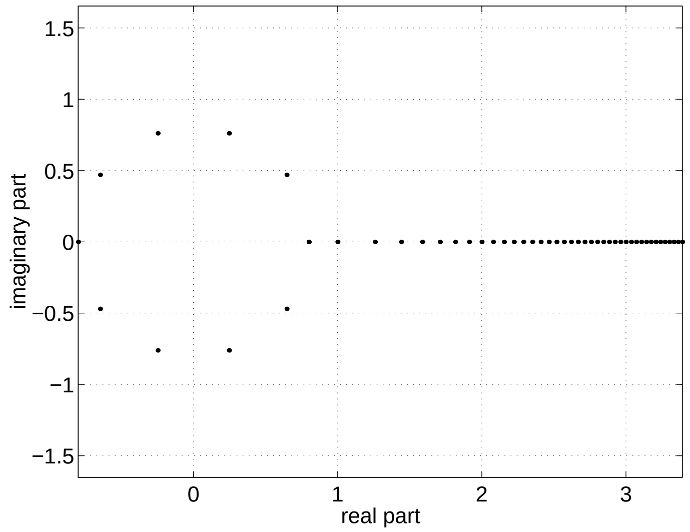

Figure 2: The spectrum of the matrix $\boldsymbol{A}$ for the clustered normal test for the case where the dimension of the matrix $\boldsymbol{A}$ is $n = 50$, the radius of the circle is $r = 0.8$ and the number of different eigenvalues on the radius of the circle is $n_1 = 10$.

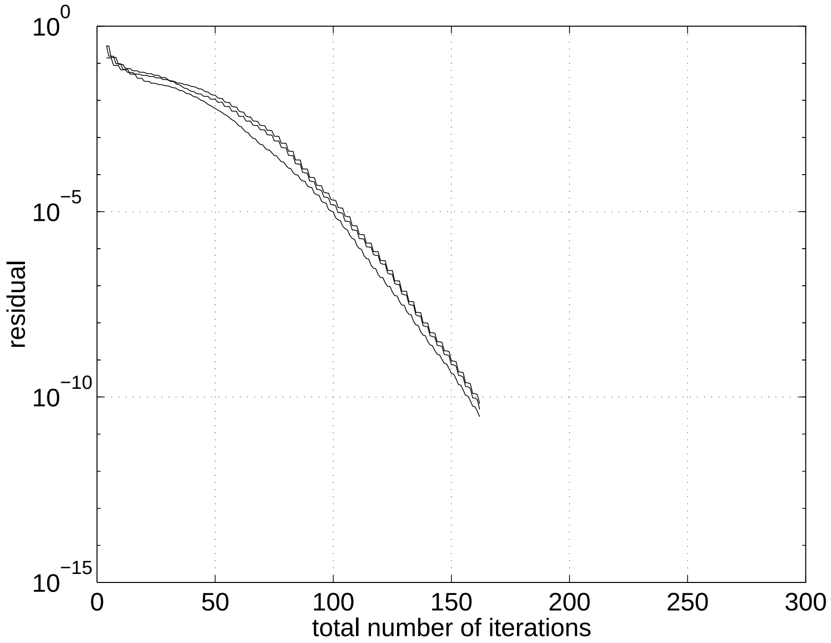

Figure 3: Test results for the band algorithm applied to the non-normal test solved for three right-hand sides. The dimension of the matrix $\boldsymbol{A}$ is $n = 2500$; the parameter values are $p = 0.2$ and $q = 3$.

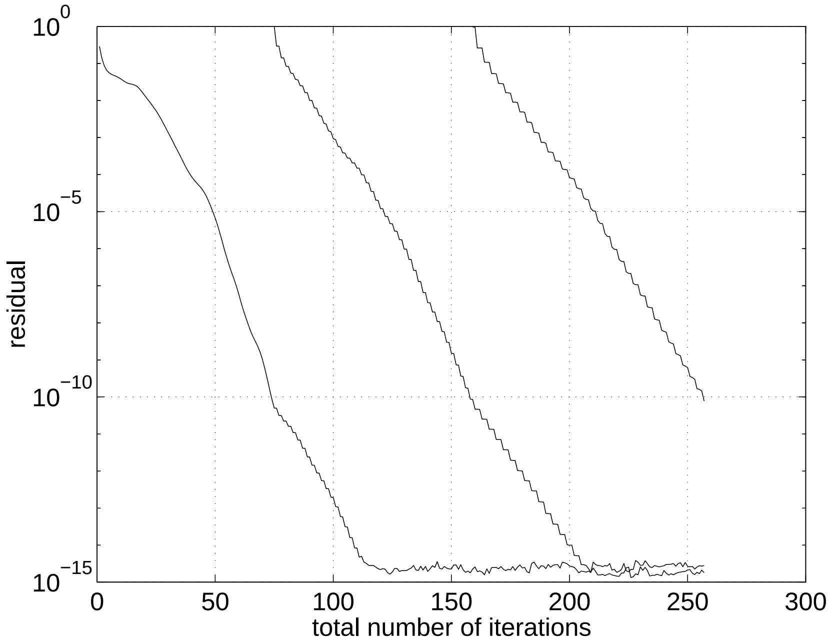

Figure 4: Test results for staircase algorithm applied to the non-normal test solved for three right-hand sides. The dimension of the matrix $\boldsymbol{A}$ is $n = 2500$; the parameter values are $p = 0.2$ and $q = 3$.

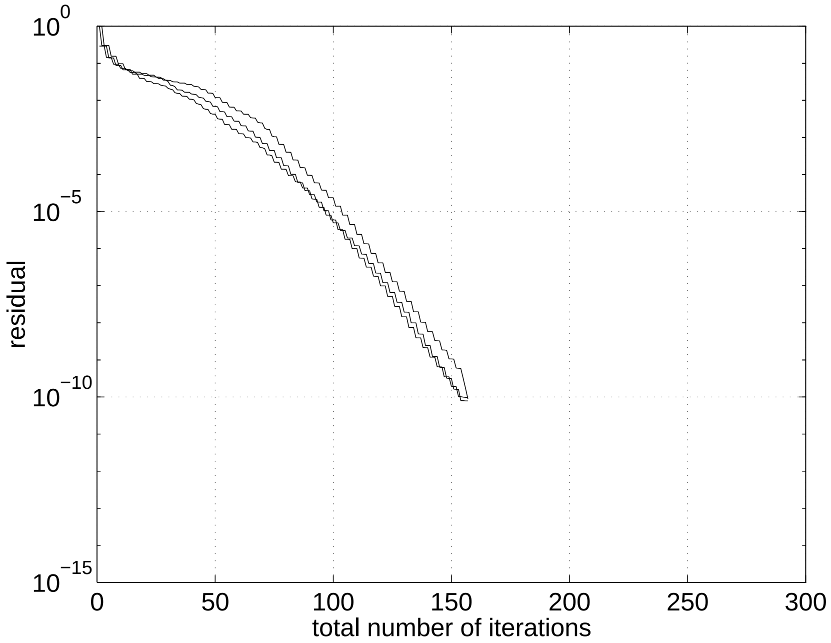

Figure 5: Test results for the adaptive band algorithm applied to the non-normal test solved for three right-hand sides. The dimension of the matrix $\boldsymbol{A}$ is $n = 2500$; the parameter values are $p = 0.2$ and $q = 3$.

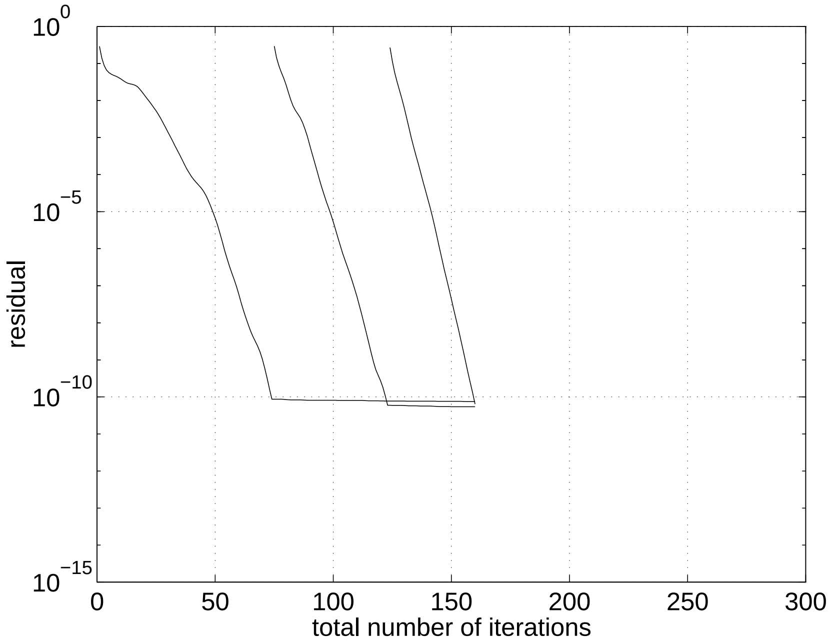

Figure 6: Test results for adaptive staircase algorithm applied to the non-normal test solved for three right-hand sides. The dimension of the matrix $\boldsymbol{A}$ is $n = 2500$; the parameter values are $p = 0.2$ and $q = 3$.

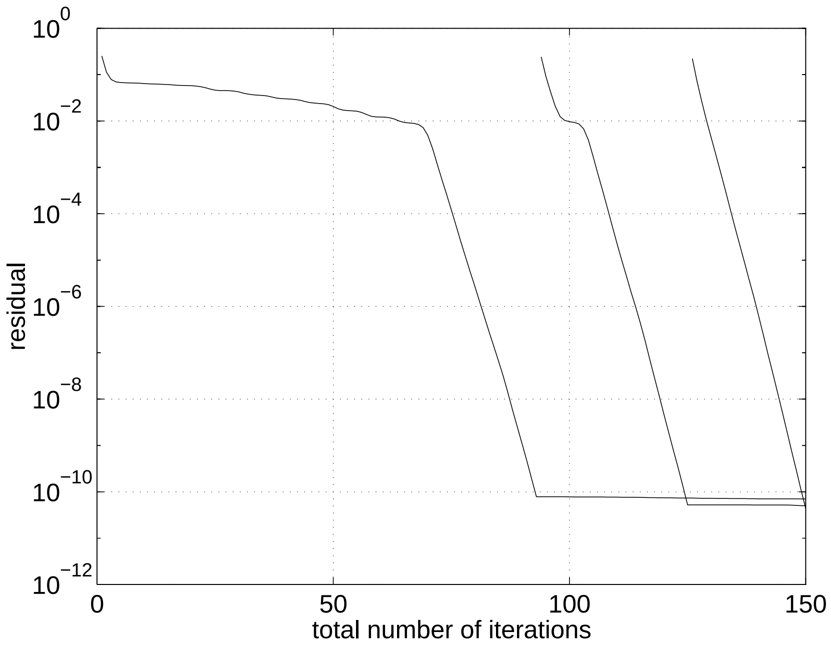

Figure 7: Test results for adaptive staircase algorithm applied to the clustered normal test solved for three right-hand sides. The dimension of the matrix $\boldsymbol{A}$ is $n = 2500$; the parameter values are $n_1 = 10$, $r = 0.1$ and $q = 3$.

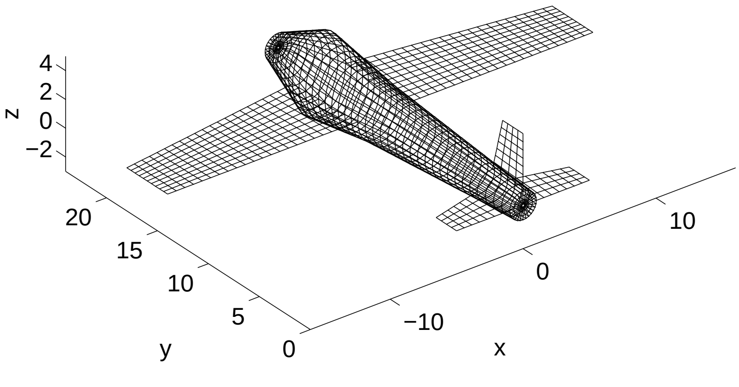

Figure 8: Wire model of an airplane. This model is used to generate a dense test matrix for the CEM test, by integral equations for induced currents.

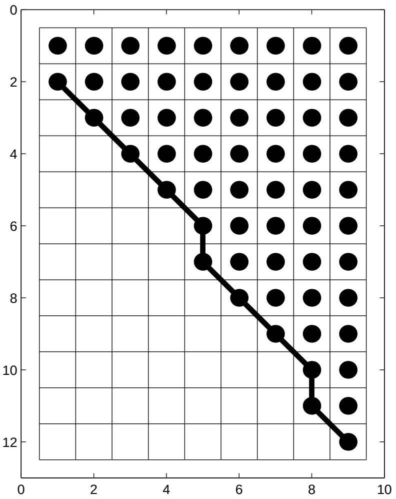

Figure 9: Typical nonzero structure for the matrix $\boldsymbol{H}$ generated by the staircase algorithm and the adaptive staircase algorithm. This particular graph is generated from test results on the adaptive staircase algorithm applied to the non-normal test solved for 3 right-hand sides. The residual norms are smaller than 10^{-1}. The dimension of the matrix $\boldsymbol{A}$ is $n = 30$; the parameter values are $p = 0.05$ and $q = 3$. Note how the bandwidth increases at two different points, when new right-hand sides are incorporated into the basis.

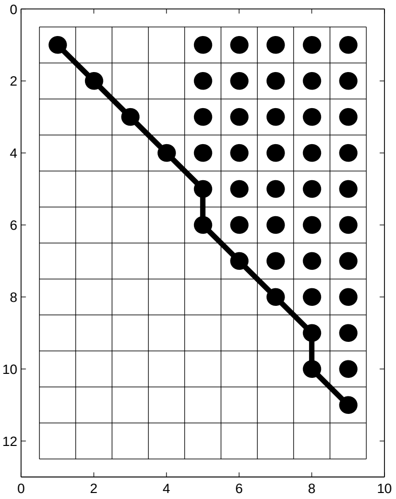

Figure 10: Typical nonzero structure for the matrix $\boldsymbol{T}$ generated by the adaptive staircase algorithm. This particular graph is generated from test results on the adaptive staircase algorithm applied to the non-normal test solved for 3 right-hand sides. The residual norms are smaller than 10^{-1}. The dimension of the matrix $\boldsymbol{A}$ is $n = 30$; the parameter values are $p = 0.05$ and $q = 3$. Note how the bandwidth increases at two different points, when the new right-hand sides are incorporated in the basis.

References

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK Users' Guide.* SIAM, Philadelphia, second edition, 1995.

[2] W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9:17–29, 1951.

[3] Tony F. Chan and W. L. Wan. Analysis of Projection Methods for Solving Linear Systems with Multiple Right-Hand Sides. *SIAM J. Sci. Comput.*, 18:1698–1721, 1997.

[4] J. Cullum and W. E. Donath. A block Lanczos algorithm for computing the Q algebraically largest eigenvalues and a corresponding eigenspace of large, sparse real symmetric matrices. In *Proceedings of the 1974 IEEE Conference on Decision and Control, Phoenix, AZ*, pages 505–509, 1974.

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

[6] Lawrence Livermore Laboratory. *NEC-2 Manual, Part III: User's Guide.* Availible at URL: http://www.dec.tis.net/~richesop/nec/.

[7] Lawrence Livermore Laboratory. *Numerical Electromagnetics Code (NEC) Methods of Moments Part 1 & Part 2.* Available from: U.S. Department of Commerce National Technical Information Service.

[8] D. P. O'Leary. The block conjugate gradient algorithm and related methods. *Linear Algebra Appl.*, 29:293–322, 1980.

[9] M. Papadrakakis and S. Smerou. A New Implementation of the Lanczos Method in Linear Problems. *Int. J. for Numerical Methods in Engineering*, 29:141–159, 1990.

[10] B. N. Parlett. A new look at the Lanczos algorithm for solving symmetric systems and linear equations. *Linear Algebra Appl.*, 29:323–346, 1980.

[11] A. Ruhe. Implementation aspects of band Lanczos algorithms for computation of eigenvalues of large sparse symmetric matrices. *Mathematics of Computation*, 33:680–687, 1979.

[12] Axel Ruhe. Rational Krylov algorithms for nonsymmetric eigenvalue problems, II: Matrix pairs. *Lin. Alg. Appl.*, 197/198:283–296, 1994.

[13] Axel Ruhe. The Rational Krylov algorithm for nonsymmetric eigenvalue problems. III: Complex shifts for real matrices. *BIT*, 34:165–176, 1994.

[14] Axel Ruhe. Rational Krylov, a practical algorithm for large sparse nonsymmetric matrix pencils. *SIAM J. Sci. Comp.*, 19:1535–1551, 1998.

[15] Y. Saad. On the Lanczos method for solving symmetric systems with several right hand sides. *Math. Comp.*, 48:651–662, 1987.

[16] Y. Saad. *Iterative Methods For Sparse Linear Systems.* PWS Publishing Co., 1996.

[17] Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7:856–869, 1986.

[18] Bernhard A. Schmitt and Rüdiger Weiner. Matrix-free W-methods using a multiple Arnoldi iteration. *Applied Numerical Mathematics*, 18:307–320, 1995.

[19] V. Simoncini and E. Gallopoulos. An Iterative Method for Nonsymmetric Systems with Multiple Right-Hand Sides. *SIAM J. Sci. Comput.*, 16:917–933, 1995.

[20] F. Smith, Charles, Andrew F. Petterson, and Raj Mittra. A Conjugate Gradient Algorithm for the Treatment of Multiple Incident Electromagnetic Fields. *IEEE Transactions on Antennas and Propagations*, 37:1490–1493, 1989.