For the convergence factor in Lemma 1.21 we get with $k = r - 1$

$$0 \leq \min_{P_{r-1} \in \mathcal{P}_{r-1}} \max_{1 \leq i \leq n} \left[1 - \lambda_i P_{r-1}(\lambda_i)\right]^2 \leq \max_{1 \leq i \leq n} \left[1 - \lambda_i \widetilde{P}_{r-1}(\lambda_i)\right]^2 = \max_{1 \leq i \leq n} Q_r(\lambda_i)^2 = 0. \tag{90}$$

Thus, from Lemma 1.21 we get $\|x_* - x_r\|_A^2 = 0$. □

A similar behavior can be expected if A has clustered eigenvalues, see Fig. 8.

There also holds a similar theorem as Th. 1.11 for the SD method, but only the square root of the condition number occurs in the upper bound:

Theorem 1.23 (Proof [1]). *The CG method applied to the SPD system $Ax = b$ yields the following estimate for the error $\varepsilon_k := x_* - x_k$ in the energy norm*

$$\|\varepsilon_k\|_A \leq 2\Big(\frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1}\Big)^k \|\varepsilon_0\|_A . \tag{91}$$

□

Although this is an improvement over Th. 1.11 for the steepest descent method, Th. 1.23 might be still too pessimistic. A major difference over SD is the fact that convergence of CG depends on the whole spectrum of A (not only the extreme eigenvalues). In fact, as Th. 1.22 shows, the CG method converges in at most r steps if A has $r < n$ distinct eigenvalues. Similar behavior occurs if there are clusters of eigenvalues, see Fig. 8.

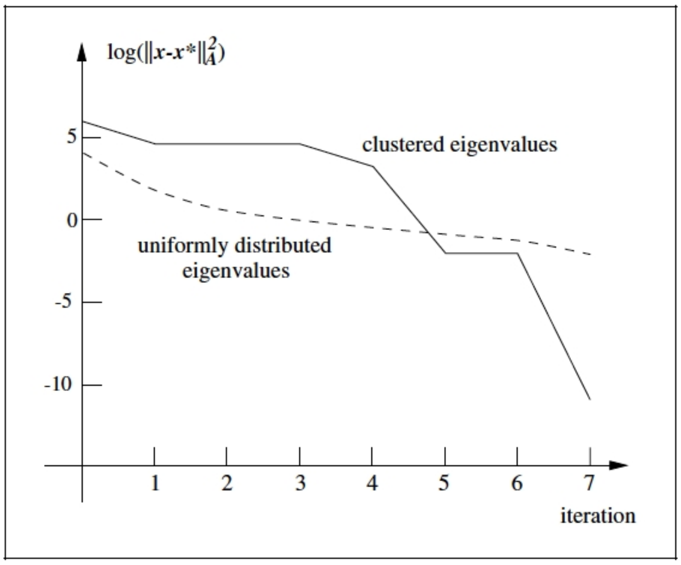

Figure 8: Convergence example of CG for uniformly distributed and clustered spectrum, taken from [11].

1.6 Krylov subspace methods

1.6.1 Krylov subspace projection

We now turn to the general case where $A \in \mathbb{R}^{n \times n}$ is an arbitrary nonsingular matrix.

The basic concept that we will use is a projection property called *Petrov-Galerkin orthogonality*: Assume we have two subspaces of $\mathbb{R}^n$ that we denote with $\mathcal{K}$ and $\mathcal{L}$. These linear subspaces might be of reduced dimensionality, e.g. $m < n$. Starting from an initial vector $x_0 \in \mathbb{R}^n$ we want to define an updated approximation $\widetilde{x} = x_0 + \delta$ with update δ in the space $\mathcal{K}$ in such a way that the new residual is perpendicular to the (test) space $\mathcal{L}$, that is:

$$\text{Find} \quad \widetilde{x} = x_0 + \delta,\, \delta \in \mathcal{K} \quad \text{such that } \, b - A\widetilde{x} = r_0 - A\delta \perp \mathcal{L}. \tag{92}$$

Fig. 9 shows this process of subspace projection.

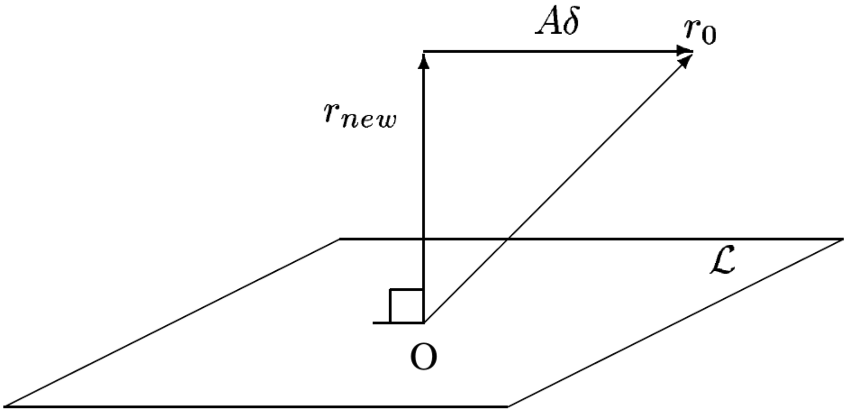

Figure 9: Subspace projection, taken from [1].

Krylov subspace methods aim at finding a good approximation x_m of the solution of the linear system $Ax = b$ in the Krylov subspace $\mathcal{K}_m$ $(m \ll n)$, i.e., $x_m \in x_0 + \mathcal{K}_m$ which satisfies the orthogonality

$$v^T(b - Ax_m) = 0 \quad \text{for all } v \in \mathcal{L}_m, \tag{93}$$

where $\mathcal{L}_m$ is called the *test space*, while $\mathcal{K}_m = \mathcal{K}_m(A, r_0)$ is the *ansatz* or *solution space*. Krylov subspace methods are linear model reduction methods for the problem $Ax = b$ that aim at reducing computational effort by utilizing the lower dimensionality of the involved subspaces. In fact, we will see that the computational effort of one projection step (iteration) is reduced to the solution of a system of smaller size.

The different Krylov subspace methods differ in the choice of the test space $\mathcal{L}_m$. There are three typical choices:

1. $\mathcal{L}_m = \mathcal{K}_m$ (*'Full orthogonalization methods' e.g. CG, D-Lanczos*)

$$r_m \perp \mathcal{K}_m \quad (\Leftrightarrow A\varepsilon_m \perp \mathcal{K}_m). \tag{94}$$

   For symmetric A

$$\varepsilon_m \perp A\mathcal{K}_m \quad (\varepsilon_m \perp_A \mathcal{K}_m \quad \text{if } A \text{ SPD}). \tag{95}$$

   In the SPD case the orthogonality requirement has a unique solution satisfying [1]

$$\|x_m - x_*\|_A = \min_{x \in x_0 + \mathcal{K}_m(A, r_0)} \|x - x_*\|_A. \tag{96}$$

2. $\mathcal{L}_m = A\mathcal{K}_m$ (*'Generalized minimal residual methods': GMRES*)

$$r_m \perp A\mathcal{K}_m \quad (\Leftrightarrow A\varepsilon_m \perp A\mathcal{K}_m). \tag{97}$$

3. $\mathcal{L}_m = \mathcal{K}_m(A^T, \widetilde{r}_0)$ for some $\widetilde{r}_0$ (*Biorthogonal methods, e.g. BiCG*).

Our focus here shall be on the GMRES method ($\mathcal{L}_m = A\mathcal{K}_m$).

Lemma 1.24. *The orthogonality* (97) *is equivalent to*

$$\|r_m\|_2 = \|b - Ax_m\|_2 = \min_{x \in x_0 + \mathcal{K}_m(A, r_0)} \|b - Ax\|_2, \tag{98}$$

which motivates the term minimal residual method.

Proof. To make (98) plausible imagine an arbitrary $x \in x_0 + \mathcal{K}_m$, where x_0 and hence r_0 are fixed. The norm of a related (generic) residual vector $r \in r_0 + A\mathcal{K}_m = r_0 + \mathcal{L}_m$ is indeed minimized iff r is perpendicular to $\mathcal{L}_m = A\mathcal{K}_m$: Write $r = r_0 + \ell = (r_0^{\|_{\mathcal{L}_m}} + \ell) + r_0^{\perp_{\mathcal{L}_m}}$, $\ell \in \mathcal{L}_m$ (the superscript $\|_{\mathcal{L}_m}$ and $\perp_{\mathcal{L}_m}$ denote parallel and perpendicular component with respect to $\mathcal{L}_m$, respectively) then $\|r\|_2^2 = \|r_0^{\|_{\mathcal{L}_m}} + \ell\|_2^2 + \|r_0^{\perp_{\mathcal{L}_m}}\|_2^2$ is minimized for $\ell = -r_0^{\|_{\mathcal{L}_m}}$. Hence, a minimal residual yields $r = r_0 - r_0^{\|_{\mathcal{L}_m}} = r_0^{\perp_{\mathcal{L}_m}} \in \mathcal{L}_m^{\perp}$ and also conversely, r perpendicular to the test space $\mathcal{L}_m$ yields a minimal residual norm. □

The characterization (98) as minimization problem would suggest to calculate the iterate x_m with the ansatz

$$x_m = x_0 + K_m c, \tag{99}$$

where $K_m = [r_0|Ar_0|\cdots|A^{m-1}r_0] \in \mathbb{R}^{n\times m}$ is the Krylov matrix and $c \in \mathbb{R}^m$ the corresponding coefficient vector. From the ansatz (99) one gets $r_m = r_0 - AK_m c$ and hence, the minimum condition (98) yields

$$c = \underset{d \in \mathbb{R}^m}{\arg\min} \|r_0 - AK_m d\|_2. \tag{100}$$

In general, this least squares problem will be ill-conditioned (nearly singular Krylov matrix), and hence the method will be numerically unstable. This is the reason why orthonormalization procedures for the Krylov spaces have to be incorporated into the methods.

Definition 1.25 (Steps in a Krylov subspace method). *A Krylov subspace algorithm consists of the following steps:*

- *Construct the (Arnoldi) matrix $V_m \in \mathbb{R}^{n\times m}$ whose columns form an orthonormal basis (i.e. $V_m^T V_m = I_m$) of the Krylov space $\mathcal{K}_m$. Compare with the forthcoming Ch. 1.6.2.*
- *Construct the matrix $W_m \in \mathbb{R}^{n\times m}$ whose columns form a basis of the test space $\mathcal{L}_m$. For $\mathcal{L}_m = A\mathcal{K}_m$ this is simply $W_m = AV_m$, which is then computed in parallel in step one.*
- *Make the ansatz $x_m = x_0 + V_m y_m$, where $y_m \in \mathbb{R}^m$ is the vector of weights to be determined.*
- *Require the Petrov-Galerkin orthogonality*

$$W_m^T(Ax_m - b) = 0 \Leftrightarrow W_m^T A V_m y_m = W_m^T r_0. \tag{101}$$

The iterate takes the form

$$x_m = x_0 + V_m (W_m^T A V_m)^{-1} W_m^T r_0 \tag{102}$$

Note that the matrix $W_m^T A V_m$ is only of size $m \times m$, which is OK for $m \ll n$.

Remark 1.26. Also note that for $W_m = AV_m$ (GMRES) the linear system (101) has the form $W_m^T W_m y_m = W_m^T r_0$, which are the normal equations of the linear least squares problem

$$y_m = \underset{y \in \mathbb{R}^m}{\arg\min} \|r_0 - AV_m y\|_2. \tag{103}$$

Note that from the ansatz $x_m = x_0 + V_m y_m$ one gets $Ax_m = Ax_0 + AV_m y_m \approx Ax_0 + r_0 = b$. □

In the following we will describe the involved orthogonalization procedure (Arnoldi/Lanczos) and the computational details of the generalized minimal residual (GMRES) method in a little more detail.

1.6.2 The Arnoldi and Lanczos Procedure: Orthogonalization of $\mathcal{K}_m$

We consider an arbitrary matrix $A \in \mathbb{R}^{n\times n}$ and a given residual vector $r_0 \in \mathbb{R}^n$ with the corresponding Krylov subspace ($m \le n$)

$$\mathcal{K}_m = \mathcal{K}_m(A, r_0) = \operatorname{span}\{r_0, Ar_0, \ldots, A^{m-1}r_0\}. \tag{104}$$

Let the matrix

$$K_m = [r_0|Ar_0|\cdots|A^{m-1}r_0] \in \mathbb{R}^{n\times m} \tag{105}$$

the corresponding *Krylov matrix.*

The Arnoldi procedure is based on the Gram-Schmidt algorithm to construct an orthonormal set/basis of the column vectors of K_m. The basic idea is to start with $v_1 = r_0/\|r_0\|_2$, then in the j-th step multiply the current Arnoldi vector v_j with A and orthonormalize Av_j against all previous Arnoldi vectors. If the plain Gram-Schmidt procedure is used for orthonormalization, this leads to the following steps

$$\begin{aligned}
& v_1 = r_0/\|r_0\|_2 \\
w_1 = Av_1 - (v_1^T Av_1)v_1, \quad & v_2 = w_1/\|w_1\|_2 \\
w_2 = Av_2 - (v_1^T Av_2)v_1 - (v_2^T Av_2)v_2, \quad & v_3 = w_2/\|w_2\|_2 \\
& \vdots \\
w_m = Av_m - (v_1^T Av_m)v_1 - \ldots - (v_m^T Av_m)v_m, \quad & v_{m+1} = w_m/\|w_m\|_2
\end{aligned} \tag{106}$$

This generates the Arnoldi matrix

$$V_m = [v_1|v_2|\cdots|v_m] \in \mathbb{R}^{n\times m} \tag{107}$$

and the upper Hessenberg matrix $\hat{H}_m \in \mathbb{R}^{(m+1)\times m}$

$$\hat{H}_m = \begin{bmatrix} h_{11} & h_{12} & h_{13} & \ldots & h_{1m} \\ h_{21} & h_{22} & h_{23} & \ldots & h_{2m} \\ & h_{32} & h_{33} & & h_{3m} \\ & & \ddots & \ddots & \vdots \\ & & & h_{m(m-1)} & h_{mm} \\ & & & & h_{(m+1)m} \end{bmatrix} = \begin{bmatrix} v_1^T Av_1 & v_1^T Av_2 & v_1^T Av_3 & \ldots & v_1^T Av_m \\ v_2^T Av_1 & v_2^T Av_2 & v_2^T Av_3 & \ldots & v_2^T Av_m \\ & v_3^T Av_2 & v_3^T Av_3 & & v_3^T Av_m \\ & & \ddots & \ddots & \vdots \\ & & & v_m^T Av_{m-1} & v_m^T Av_m \\ & & & & v_{m+1}^T Av_m \end{bmatrix}.$$

Note that the off-diagonal elements of $\hat{H}_m$ are calculated in the normalization step of (106) by calculating $\|w_m\|_2 = v_{m+1}^T w_m = v_{m+1}^T Av_m$ using the mutual orthogonality of the Arnoldi vectors, i.e., $v_{m+1}^T v_j$, $j = 1, \ldots, m$.

If a breakdown occurs in the m-th step, $w_m = 0$ is defined but not v_{m+1} and so the algorithm stops; in this case the element $h_{m+1,m} = 0$.

Lemma 1.27. *If the Arnoldi iteration* (106) *does not terminate prematurely, the vectors $v_1, \ldots, v_m$ form an orthonormal basis of the Krylov space $\mathcal{K}_m$, i.e., $V_m^T V_m = I_m$. Further, the operator $P_m = V_m V_m^T$ is the orthogonal projection onto $\mathcal{K}_m$.* □

In any case, we have from (106) that

$$Av_j = w_j + \sum_{i=1}^{j} h_{ij} v_i, \tag{108}$$

which means in matrix notation [15]

[15] Note for the last identity that $V_{m+1}\hat{H}_m = [V_m|v_{m+1}] \begin{bmatrix} H_m \\ (v_{m+1}^T Av_m)e_m^T \end{bmatrix} = V_m H_m + (v_{m+1}^T Av_m)v_{m+1}e_m^T$ and due to orthogonality $v_{m+1}^T Av_m = \|w_m\|_2$ (use (106)).

$$AV_m = V_m H_m + w_m e_m^T \ \ (= V_{m+1}\hat{H}_m), \tag{109}$$

with $H_m \in \mathbb{R}^{m\times m}$ is the square Hessenberg matrix obtained from $\hat{H}_m$ by removing the last row and e_m denotes the m-th unit vector.

Lemma 1.28 (Arnoldi identity). *From* (109) *we conclude that*

$$V_m^T AV_m = H_m + \underbrace{V_m^T w_m}_{=0} e_m^T = H_m. \tag{110}$$

Proof. We have $V_m^T AV_m = H_m + \underbrace{V_m^T w_m}_{=0} e_m^T = H_m$. □

Theorem 1.29 ([3]). *The Arnoldi procedure generates a reduced QR-decomposition of the Krylov matrix K_m in the form*

$$K_m = V_m R_m \tag{111}$$

with the Arnoldi matrix V_m from (107) *and a triangular matrix $R_m \in \mathbb{R}^{m\times m}$. Further, the Hessenberg matrix H_m is an projection of A onto the Krylov subspace spanned by the columns of K_m, i.e.,*

$$H_m = V_m^T AV_m. \tag{112}$$

□

Remark 1.30. "H_m is a projected version of A": If we consider $x \in \mathcal{K}_m$ expressed in the basis V_m, i.e., $x = V_m y$, then the orthogonal projection $V_m V_m^T$ onto $\mathcal{K}_m$ of Ax is

$$V_m V_m^T Ax = V_m V_m^T AV_m y = V_m H_m y, \tag{113}$$

that is, the coefficients (or coordinates w.r.t. the basis V_m) of the projected image is given by $H_m y$.
If x is not yet in $\mathcal{K}_m$, we can simply consider the projected version $A_m := P_m AP_m = V_m H_m V_m^T$ with the projection $P_m = V_m V_m^T$: $A_m x = V_m H_m(V_m^T x) = V_m H_m y$, where $y = V_m^T x$ are the coordinates of the projection image of x, i.e., $P_m x = V_m(V_m^T x) = V_m y$, w.r.t. the basis V_m. □

Remark 1.31. Since H_m is a projection of A the eigenvalues of H_m are good estimates for (some of) those of A. They are called Arnoldi estimates or Ritz values, see Ch. **??**. □

Remark 1.32. The Arnoldi procedure is connected with polynomial approximation. The problem of finding a monic polynomial p_m of degree m [16] such that the norm $\|p_m(A)r_0\|_2$ becomes minimal, has a unique solution in the case where K_m has full rank, namely the characteristic polynomial of H_m, i.e., $\chi(H_m)$. Further, for $x \in \mathcal{K}_m$ one can show

$$V_m \chi(H_m) V_m^T x = \chi(A)x = 0. \tag{114}$$

This shows that $V_m \chi(H_m) V_m^T$ behaves like the characteristic polynomial of A when restricted to $\mathcal{K}_m$. Therefore the eigenvalues of H_m (Ritz values) can be expected to be good approximations to some of the eigenvalues of A. In fact, one can show that each eigenvalue of H_m is also an eigenvalue of $A_m := V_m H_m V_m^T = P_m AP_m$ with the projection $P_m = V_m V_m^T$ and all other eigenvalues of A_m are zero. □

The Arnoldi procedure is usually implemented in a numerically stable way known as *Modified Gram-Schmidt algorithm* (MGS).
The Arnoldi iteration in the modified Gram-Schmidt variant is summarized in Alg. 5.

[16] A polynomial where the highest coefficient is one, i.e., polynomials of the form $p_m(z) = c_0 + c_1 z + \ldots + c_{m-1}z^{m-1} + z^m$.

Algorithm 5: Arnoldi iteration (Modified Gram-Schmidt variant)

```
Data: Matrix $A$, vector $r_0$, number $m$
Result: Arnoldi vectors $v_1, \ldots, v_m$, Upper Hessenberg matrix $\hat{H}_m = (h_{ij})$
Initialization: $v_1 \leftarrow r_0 / \|r_0\|_2$
for $j = 1 \ldots m$ do
    $w_j \leftarrow A v_j$
    for $i = 1 \ldots j$ do
        $h_{ij} \leftarrow v_i^T w_j$
        $w_j \leftarrow w_j - h_{ij} v_i$
    end
    $h_{(j+1),j} \leftarrow \|w_j\|_2$
    if $h_{j+1,j} = 0$ then
        Stop
    end
    $v_{j+1} \leftarrow w_j / h_{(j+1),j}$
end
```

In the case where A is symmetric, also the upper Hessenberg matrix H_m is symmetric ($H_m = V_m^T A V_m = V_m^T A^T V_m = H_m^T$) and therefore a symmetric tridiagonal matrix

$$T_m = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_m \\ & & & \beta_m & \alpha_m \end{bmatrix}.$$

As a consequence, the inner loop in Alg. 5 simplifies, see Alg. 6. This is called the *Lanczos iteration.*

Algorithm 6: Lanczos iteration

```
Data: Matrix $A$ symmetric, vector $r_0$, number $m$
Result: Arnoldi vectors $v_1, \ldots, v_m$, tridiagonal matrix $T_m$
Initialization: $\beta_1 \leftarrow 0$, $v_0 \leftarrow 0, v_1 \leftarrow r_0 / \|r_0\|_2$
for $j = 1 \ldots m$ do
    $w_j \leftarrow A v_j - \beta_j v_{j-1}$
    $\alpha_j \leftarrow v_j^T w_j$
    $w_j \leftarrow w_j - \alpha_j v_j$
    $\beta_{j+1} \leftarrow \|w_j\|_2$
    if $\beta_{j+1} = 0$ then
        Stop
    end
    $v_{j+1} \leftarrow w_j / \beta_{j+1}$
end
```

1.6.3 Generalized minimal residual method (GMRES)

Here $A \in \mathbb{R}^{n \times n}$ is a (nonsingular) square matrix. As described in the beginning of this chapter the GMRES method requires the Petrov-Galerkin orthogonality $r_m \perp A\mathcal{K}_m$. This leads to the normal equations (see (101))

$$W_m^T W_m y_m = W_m^T r_0, \tag{115}$$

for $W_m = AV_m$ and the ansatz for the approximate solution $x_m = x_0 + V_m y_m$. The normal equations (115) can be simplified in a practically more convenient way:

Lemma 1.33. *The normal equations* (115) *in the computation of a GMRES iterate can be reformulated as*

$$\hat{H}_m^T \hat{H}_m y_m = \hat{H}_m^T \, \|r_0\|_2 \, e_1, \tag{116}$$

where e_1 *is the unit vector* $e_1 = (1, 0, 0, \ldots, 0)^T$*. Eqn.* (116) *has a unique solution under the "no-break-down assumption":* $h_{j+1,j} \neq 0, \; j = 1, \ldots, m-1$.

Proof. We recall the identity (109) which yields

$$W_m = AV_m = V_m H_m + w_m e_m^T \; = V_{m+1} \hat{H}_m. \tag{117}$$

Let us assume that the Arnoldi iteration does not break down, that is $h_{j+1,j} \neq 0, \; j = 1, \ldots, m-1$, which implies that $\hat{H}_m$ has full column rank m. Now note that

$$W_m^T W_m = \hat{H}_m^T \underbrace{V_{m+1}^T V_{m+1}}_{I_{m+1}} \hat{H}_m = \hat{H}_m^T \hat{H}_m, \tag{118}$$

is non-singular due to our "no-break-down assumption". Also the r.h.s. of (115) simplifies due to the orthogonality of $v_1, \ldots, v_{m+1}$ to

$$W_m^T r_0 = \hat{H}_m^T V_{m+1}^T r_0 = \hat{H}_m^T \underbrace{(v_1^T r_0)}_{=\|r_0\|_2} \, e_1 \; = \hat{H}_m^T \, \|r_0\|_2 \, e_1. \tag{119}$$

Thus, the resulting normal equations in the GMRES method take the form

$$\hat{H}_m^T \hat{H}_m y_m = \hat{H}_m^T \, \|r_0\|_2 \, e_1. \tag{120}$$

They are uniquely solvable [17] under the "no-break-down assumption"[18]. □

The normal equations (116) are equivalent to the linear least squares problem

$$y_m = \arg\min_{y \in \mathbb{R}^m} \left\| e_1 \, \|r_0\|_2 - \hat{H}_m y \right\|_2. \tag{121}$$

The GMRES algorithm is given in Alg. 7; it consists of the Arnoldi iteration followed by the solution of the linear least squares problem (121).

```
Algorithm 7: GMRES algorithm
  Data: Matrix $A \in \mathbb{R}^{n \times n}$, r.h.s. $b \in \mathbb{R}^n$, initial vector $x_0 \in \mathbb{R}^n$
  Result: $x \in \mathbb{R}^n$ approximate solution
  Initialization: $m \leftarrow 1, r_0 \leftarrow b - Ax_0$
  while convergence criterion not satisfied do
          Compute $V_m, \hat{H}_m$ from Arnoldi iteration Alg. 5
          $y_m \leftarrow \arg\min_{y \in \mathbb{R}^m} \|e_1 \|r_0\|_2 - \hat{H}_m y\|_2$        (Linear least squares problem)
          $x_m \leftarrow x_0 + V_m y_m$
          $m \leftarrow m + 1$
  end
  $x \leftarrow x_{m-1}$
```

Remark 1.34. (Efficient Updates) Note that in Alg. 7 the new Arnoldi vector v_m is obtained from the previously computed $v_1, \ldots, v_{m-1}$. Also the matrix $\hat{H}_m$ is only updated form the previously computed $\hat{H}_{m-1}$ by adding a column and the element $h_{m+1,m}$. Similarly, the QR-decomposition of $\hat{H}_m$, which is used for solving the linear least squares problem, can be efficiently updated. □

[17] Because of the Hessenberg structure of $\hat{H}_m$ the QR-decomposition is a good choice to solve the underlying linear least squares problem, rather than applying the more expensive Cholesky-decomposition to the normal equations.

[18] The system is solvable until a break-down occurs.

Remark 1.35. **(Restarts)** To control the required memory the GMRES algorithm is usually restarted after a maximum dimension $m = m_{max}$ is reached. In this *restarted version* the approximation $x_{m_{max}}$ is used as the new starting vector while all the previously computed information is deleted. □

Remark 1.36. **(Truncated (Quasi) GMRES)** An incomplete version of GMRES is derived by incomplete orthogonalization. Only the k previous Arnoldi vectors are kept in memory (similar idea as in *limited memory* quasi Newton methods in numerical optimization). For further details see [1, 11]. □

Example 1.37. Consider the Poisson equation in three dimensions on $\Omega = (-a, a)^3$

$$\begin{aligned} -\Delta u &= f \equiv 1 \\ u_{|\partial\Omega} &= 0, \end{aligned} \tag{122}$$

and a finite difference discretization with uniform mesh size $h = 2a/30$ with resulting sparse system matrix

$$A = K_{1d} \otimes I \otimes I + I \otimes K_{1d} \otimes I + I \otimes I \otimes K_{1d} \in \mathbb{R}^{30^3 \times 30^3}. \tag{123}$$

Fig. 10 shows the performance of GMRES for different maximum Krylov space dimension and the CG method (as benchmark, since it is the ideal method for this SPD system).

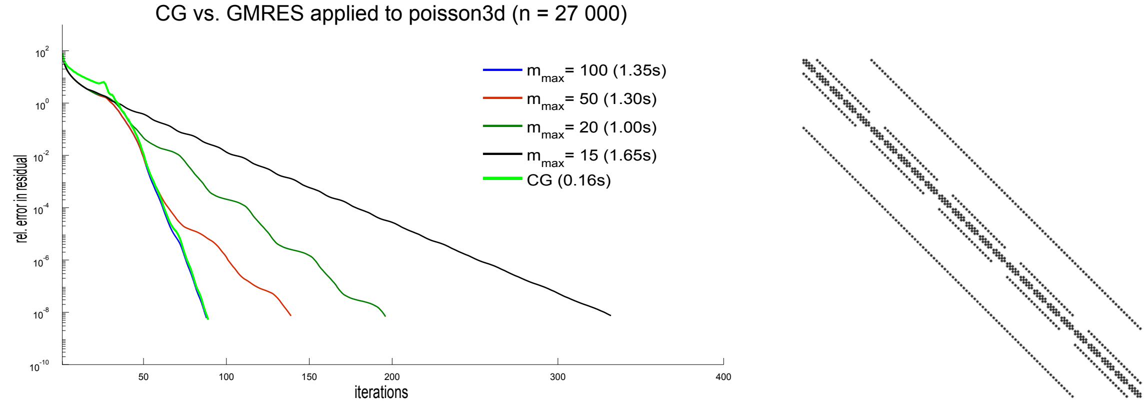

Figure 10: GMRES with different maximum Krylov space dimension and the CG method applied to (122). Right: Sparsity pattern of A.

On convergence of GMRES.

The following convergence result for GMRES is obtained from polynomial representation of the Krylov space $\mathcal{K}_m = \{x_0 + p(A)r_0 : p \in \mathcal{P}_{m-1}\}$ and the rather restrictive assumption of diagonalizable A.

Theorem 1.38 (Proof [3]). *Let A be diagonalizable: $A = X\Lambda X^{-1}$, with $\Lambda = diag(\lambda_1, \ldots, \lambda_n)$ the diagonal matrix of eigenvalues. Let further* [19]

$$\epsilon_m := \min_{q \in \mathcal{P}_m,\, q(0)=1} \max_{i=1,\ldots,n} |q(\lambda_i)|. \tag{124}$$

Then there holds for the 2-norm of the m-th residual

$$\|r_m\|_2 \leq \epsilon_m \, \kappa_2(X) \, \|r_0\|_2. \tag{125}$$

Remark 1.39. Note that the condition number $\kappa_2(X)$ can be large even if that of A is small, e.g. if A is close to non-diagonalizable. Also, X is not known in practice, so this result is of limited practical value.

Preconditioning does not have such a clear positive effect as in the CG method, e.g. can also have adverse effects on $\kappa_2(X)$. However, 'bunching eigenvalues' is still a valid strategy since ϵ_m can be bounded by a complex analogue of the Chebyshev min-max theorem [1].

[19] Note that the error of the m-th iteration has a representation $\varepsilon_m = \varepsilon_0 + p_{m-1}(A)r_0 = (I - p_{m-1}(A)A)\,\varepsilon_0 =: q_m(A)\varepsilon_0$ with $q_m \in \mathcal{P}_m$ and $q_m(0) = 1$.