

Chapter 6

Krylov Subspace Methods that are Based on a Projection of the Residual

Remark 6.1. Idea. The methods presented in this section determine the iterate $\underline{x}^{(k)}$ at the manifold $\underline{x}^{(0)} + K_k(\underline{r}^{(0)}, A)$ such that the corresponding residual $\underline{r}^{(k)}$ is orthogonal to $K_k(\underline{r}^{(0)}, A)$. That means, $\underline{r}^{(k)}$ is projected in the orthogonal complement $K_k(\underline{r}^{(0)}, A)^\perp$ of $K_k(\underline{r}^{(0)}, A)$.

Orthogonality with respect to a subspace is already known as a feature of the best approximation problem in pre Hilbert spaces. There, the error is orthogonal to the subspace, see Numerical Mathematics I. In iterative schemes for solving linear equations, the error is not known. Here, one tries to construct an efficient method by projecting the residual vector. \square

6.1 General Matrices

Remark 6.2. Full orthogonalization method. Let $Q_k = \{\underline{q}_1, \dots, \underline{q}_k\}$ be an orthonormal basis of $K_k(\underline{r}^{(0)}, A)$ computed with Arnoldi's method, Algorithm 5.3. It is $\underline{q}_1 = \underline{r}^{(0)} / \|\underline{r}^{(0)}\|_2$. Set $\beta = \|\underline{r}^{(0)}\|_2$. By the orthogonality of the columns of Q_k , it follows that

$$Q_k^T \underline{r}^{(0)} = \beta Q_k^T \underline{q}_1 = \beta \underline{e}_1. \quad (6.1)$$

Then, one finds that the iterate $\underline{x}^{(k)}$ is given by

$$\underline{x}^{(k)} = \underline{x}^{(0)} + Q_k \underline{y}_k \quad \text{with} \quad \underline{y}_k = \tilde{H}_k^{-1}(\beta \underline{e}_1), \quad (6.2)$$

where \tilde{H}_k is defined in (5.7), since the orthogonal projection of $\underline{r}^{(k)}$ in $K_k(\underline{r}^{(0)}, A)^\perp$ is unique and the iterate (6.2) fulfills $\underline{r}^{(k)} \perp K_k(\underline{r}^{(0)}, A)$:

$$\begin{aligned} Q_k^T \underline{r}^{(k)} &= Q_k^T (\underline{b} - A \underline{x}^{(k)}) = Q_k^T (\underline{b} - A \underline{x}^{(0)} - \beta A Q_k \tilde{H}_k^{-1} \underline{e}_1) \\ &= Q_k^T \underline{r}^{(0)} - \beta \underbrace{Q_k^T A Q_k}_{=\tilde{H}_k} \tilde{H}_k^{-1} \underline{e}_1 = Q_k^T \underline{r}^{(0)} - \beta \underline{e}_1 = \underline{0}, \end{aligned}$$

where (6.2), (5.7), and (6.1) have been used.

The algorithm which is based on this approach is called full orthogonalization method (FOM). Since it is of little relevance in practice, it will not be presented here in detail. Similarly to GMRES, an early break down of the Arnoldi process is equivalent of already having computed the solution. FOM possesses the same fundamental problem as GMRES since the whole basis of the $K_k(\underline{r}^{(0)}, A)$ has to be stored. In contrast

to GMRES, the iterate of FOM is undefined if \tilde{H}_k is singular. This situation can happen, e.g., if A is a symmetric indefinite matrix. \square

6.2 Symmetric Matrices

Remark 6.3. SYMMLQ for symmetric matrices. If A is a symmetric matrix, there is a way to perform FOM with a short recurrence, i.e., without having to store the whole basis $\{\underline{q}_1, \dots, \underline{q}_k\}$ of $K_k(\underline{r}^{(0)}, A)$. The resulting method is called SYMMLQ. This method will not be presented here. Instead, the case that A is symmetric and positive definite will be studied in detail. Then, SYMMLQ can be simplified, leading to the famous conjugate gradient (CG) method. \square

Remark 6.4. Lanczos algorithm for a s.p.d. matrix. CG will be derived from the Lanczos method, Algorithm 5.12. Starting point is the Cholesky¹ decomposition of \tilde{H}_k , which is by Lemma 5.14 a non-singular matrix,

$$\tilde{H}_k = L_k D_k L_k^T = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ l_1 & 1 & \cdots & 0 & 0 \\ & \ddots & \ddots & & \\ & & & l_{k-1} & 1 \\ 0 & 0 & \cdots & l_{k-1} & 1 \end{pmatrix} \begin{pmatrix} d_1 & 0 & \cdots & 0 & 0 \\ 0 & d_2 & \cdots & 0 & 0 \\ & & \ddots & & \\ & & & \ddots & \\ 0 & 0 & \cdots & d_k & \end{pmatrix} \begin{pmatrix} 1 & l_1 & \cdots & 0 & 0 \\ & \ddots & \ddots & & \\ & & & 0 & 0 & \cdots & 1 & l_{k-1} \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}. \quad (6.3)$$

Define $\hat{P}_k = Q_k L_k^{-T} = (\hat{\underline{p}}_1, \dots, \hat{\underline{p}}_k)$. The columns of \hat{P}_k are linear combinations of the columns of Q_k such that $\text{span}\{\hat{\underline{p}}_1, \dots, \hat{\underline{p}}_k\} \subset K_k(\underline{r}^{(0)}, A)$. Since L_k is a non-singular matrix and since the columns of Q_k form a basis of $K_k(\underline{r}^{(0)}, A)$, the product \hat{P}_k has full rank k and consequently, the columns of \hat{P}_k form a basis of $K_k(\underline{r}^{(0)}, A)$. Notice that $\hat{\underline{p}}_1 = \underline{q}_1 = \underline{r}^{(0)} / \|\underline{r}^{(0)}\|_2$. Using (6.3), one finds for the iterate (6.2) of FOM that

$$\underline{x}^{(k)} = \underline{x}^{(0)} + \beta Q_k L_k^{-T} D_k^{-1} L_k^{-1} \underline{e}_1 = \underline{x}^{(0)} + \hat{P}_k \underline{y}_k \quad (6.4)$$

with $\underline{y}_k = \beta D_k^{-1} L_k^{-1} \underline{e}_1$. \square

Lemma 6.5. Columns of \hat{P}_k are A -conjugate. *The columns $\{\hat{\underline{p}}_1, \dots, \hat{\underline{p}}_k\}$ are mutually A -conjugate, i.e., $\hat{P}_k^T A \hat{P}_k$ is a diagonal matrix.*

Proof. Using (5.7) and (6.3) gives

$$\hat{P}_k^T A \hat{P}_k = L_k^{-1} Q_k^T A Q_k L_k^{-T} = L_k^{-1} \tilde{H}_k L_k^{-T} = D_k. \quad \blacksquare$$

Remark 6.6. First version of a method. With this version of the method, it is shown that a short recurrence is possible. But this version is not yet optimal with respect to the computational costs.

The last column of \hat{P}_k is given by

$$\hat{\underline{p}}_k = \underline{q}_k - l_{k-1} \hat{\underline{p}}_{k-1}, \quad (6.5)$$

which follows immediately from $Q_k = \hat{P}_k L_k^T$. The update \underline{y}_k in (6.4) has the form $\underline{y}_k = (\underline{y}_{k-1}, \eta_k)^T$ with $\eta_k \in \mathbb{R}$, since

¹ André Louis Cholesky (1875 – 1918)

$$\begin{aligned}
\underline{y}_k &= \beta D_k^{-1} L_k^{-1} \underline{e}_1 = \beta \underbrace{\begin{pmatrix} D_{k-1} & \\ & d_k \end{pmatrix}^{-1}}_{\begin{pmatrix} D_{k-1}^{-1} & \\ & d_k^{-1} \end{pmatrix}} \underbrace{\begin{pmatrix} L_{k-1} & \underline{0} \\ \underline{0}^T & l_{k-1} & 1 \end{pmatrix}^{-1}}_{\begin{pmatrix} L_{k-1}^{-1} & \underline{0} \\ * & 1 \end{pmatrix}} \underline{e}_{1,k} \\
&= \beta \begin{pmatrix} D_{k-1}^{-1} L_{k-1}^{-1} & \underline{0} \\ * & d_k^{-1} \end{pmatrix} \underline{e}_{1,k} = \begin{pmatrix} \beta D_{k-1}^{-1} L_{k-1}^{-1} \underline{e}_{1,k-1} \\ * \end{pmatrix} = \begin{pmatrix} \underline{y}_{k-1} \\ \eta_k \end{pmatrix},
\end{aligned}$$

where $\underline{e}_{1,k}$ is the first Cartesian unit vector with k components and $\underline{e}_{1,k-1}$ the first Cartesian unit vector of length $(k-1)$. This means, the first $(k-1)$ components of \underline{y}_k are the components of \underline{y}_{k-1} . Now, one needs to find a formula for η_k .

From the definition of \underline{y}_k , it follows that $L_k D_k \underline{y}_k = \beta \underline{e}_1$, i.e.,

$$L_k \begin{pmatrix} y_{k,1} d_1 \\ y_{k,2} d_2 \\ \vdots \\ \eta_k d_k \end{pmatrix} = \begin{pmatrix} \beta \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Hence, using the form of L_k , it is $l_{k-1} y_{k,k-1} d_{k-1} + \eta_k d_k = 0$ and

$$\eta_k = -\frac{l_{k-1} y_{k,k-1} d_{k-1}}{d_k}, \quad \text{if } k \geq 2. \quad (6.6)$$

The first component η_1 is given by the first update

$$\eta_1 = \beta D_1^{-1} L_1^{-1} \underline{e}_1 = \frac{\beta}{d_1}.$$

From the representation (6.3), it follows that (notice that $\beta = \|\underline{r}^{(0)}\|_2$, α_k , $k \geq 1$, and β_k , $k \geq 2$, are the entries of H_k , compare (5.8))

$$\begin{aligned}
\alpha_1 &= d_1, \\
\beta_k &= d_{k-1} l_{k-1}, \quad k \geq 2, \\
\alpha_k &= d_{k-1} l_{k-1}^2 + d_k, \quad k \geq 2,
\end{aligned} \quad (6.7)$$

where the values α_k and β_k are available from the Lanczos method. Thus, with α_1 , the value for d_1 is given. With d_1 and β_2 , one finds l_1 . Then, with α_2, d_1, l_1 , one can compute d_2 and so on.

Inserting all terms in (6.4) gives

$$\underline{x}^{(k)} = \underline{x}^{(0)} + \hat{P}_k \underline{y}_k = \underline{x}^{(0)} + \begin{pmatrix} \hat{P}_{k-1} \hat{p}_k \\ \eta_k \end{pmatrix} \begin{pmatrix} \underline{y}_{k-1} \\ \eta_k \end{pmatrix} = \underline{x}^{(0)} + \hat{P}_{k-1} \underline{y}_{k-1} + \eta_k \hat{p}_k = \underline{x}^{(k-1)} + \eta_k \hat{p}_k. \quad (6.8)$$

Thus, the new iterate can be computed with (6.7), (6.5), and (6.6). This approach shows that a short recurrence is possible. However, it is not yet optimal from the point of view of computational costs and it can be simplified. \square

Remark 6.7. Optimal version of the method. It holds for the residual, using (6.4), that

$$\underline{r}^{(k)} = \underline{b} - A \underline{x}^{(k)} = \underline{b} - A \underline{x}^{(0)} - A \hat{P}_k \underline{y}_k = \underline{r}^{(0)} - A \underline{z}$$

with some vector $\underline{z} \in K_k(\underline{r}^{(0)}, A) = \text{span}\{\underline{q}_1, \dots, \underline{q}_k\}$ since the columns of \hat{P}_k form a basis of $K_k(\underline{r}^{(0)}, A)$, see Remark 6.4. This representation shows first that

$$\underline{r}^{(k)} \in K_{k+1}(\underline{r}^{(0)}, A) = \text{span}\{\underline{q}_1, \dots, \underline{q}_k, \underline{q}_{k+1}\}.$$

By construction, see Remark 6.2, it is also $\underline{r}^{(k)} \perp K_k(\underline{r}^{(0)}, A)$. These two properties imply that $\underline{r}^{(k)} \in \text{span}\{\underline{q}_{k+1}\}$ such that

$$\underline{r}^{(k)} = \pm \left\| \underline{r}^{(k)} \right\|_2 \underline{q}_{k+1}. \quad (6.9)$$

With (6.8), the residual vector $\underline{r}^{(k)}$ can be computed recursively by

$$\underline{r}^{(k)} = \underline{b} - A\underline{x}^{(k)} = \underline{b} - A(\underline{x}^{(k-1)} + \eta_k \hat{P}_k) = \underline{r}^{(k-1)} - \eta_k A \hat{P}_k. \quad (6.10)$$

Setting (taking the plus sign in (6.9))

$$\underline{q}_{k+1} = \frac{\underline{r}^{(k)}}{\left\| \underline{r}^{(k)} \right\|_2} \quad (6.11)$$

and denoting $\underline{p}_k = \left\| \underline{r}^{(k-1)} \right\|_2 \hat{p}_k$, one obtains with (6.10) and (6.8)

$$\underline{r}^{(k)} = \underline{r}^{(k-1)} - \frac{\eta_k}{\left\| \underline{r}^{(k-1)} \right\|_2} A \underline{p}_k = \underline{r}^{(k-1)} - \nu_k A \underline{p}_k, \quad (6.12)$$

$$\underline{x}^{(k)} = \underline{x}^{(k-1)} + \frac{\eta_k}{\left\| \underline{r}^{(k-1)} \right\|_2} \underline{p}_k = \underline{x}^{(k-1)} + \nu_k \underline{p}_k. \quad (6.13)$$

With (6.5) and (6.11), one gets for $k \geq 2$

$$\begin{aligned} \underline{p}_k &= \left\| \underline{r}^{(k-1)} \right\|_2 \hat{p}_k = \left\| \underline{r}^{(k-1)} \right\|_2 \left(\underline{q}_k - l_{k-1} \frac{\underline{p}_{k-1}}{\left\| \underline{r}^{(k-2)} \right\|_2} \right) \\ &= \underline{r}^{(k-1)} - \frac{\left\| \underline{r}^{(k-1)} \right\|_2 l_{k-1}}{\left\| \underline{r}^{(k-2)} \right\|_2} \underline{p}_{k-1} = \underline{r}^{(k-1)} + \mu_k \underline{p}_{k-1}. \end{aligned} \quad (6.14)$$

Notice that $\underline{p}_1 = \underline{r}^{(0)}$.

Now, formulas for ν_k and μ_k are needed. Multiplying (6.14) from the left-hand side with $\underline{p}_k^T A$ gives

$$\underline{p}_k^T A \underline{p}_k = \underline{p}_k^T A \underline{r}^{(k-1)} + \mu_k \left\| \underline{r}^{(k-2)} \right\|_2 \left\| \underline{r}^{(k-1)} \right\|_2 \underbrace{\hat{p}_k^T A \hat{p}_{k-1}}_{=0, \text{Lemma 6.5}} = \underline{p}_k^T A \underline{r}^{(k-1)}. \quad (6.15)$$

Multiplying (6.12) from left with $(\underline{r}^{(k-1)})^T$ and using $\underline{r}^{(j-1)} = c_j \underline{q}_j, j = 1, \dots, k+1$, see (6.9), and the orthonormality of the vectors \underline{q}_j leads to

$$\underbrace{(\underline{r}^{(k-1)})^T \underline{r}^{(k)}}_{=0} = (\underline{r}^{(k-1)})^T \underline{r}^{(k-1)} - \nu_k (\underline{r}^{(k-1)})^T A \underline{p}_k,$$

which gives together with (6.15)

$$\nu_k = \frac{\left(\underline{r}^{(k-1)}\right)^T \underline{r}^{(k-1)}}{\left(\underline{r}^{(k-1)}\right)^T A \underline{p}_k} = \frac{\left(\underline{r}^{(k-1)}\right)^T \underline{r}^{(k-1)}}{\underline{p}_k^T A \underline{p}_k}. \quad (6.16)$$

Now, multiplying (6.14) from left with $\underline{p}_{k-1}^T A$ leads to, using Lemma 6.5,

$$\underbrace{\underline{p}_{k-1}^T A \underline{p}_k}_{=0} = \underline{p}_{k-1}^T A \underline{r}^{(k-1)} + \mu_k \underline{p}_{k-1}^T A \underline{p}_{k-1},$$

which gives

$$\mu_k = -\frac{\underline{p}_{k-1}^T A \underline{r}^{(k-1)}}{\underline{p}_{k-1}^T A \underline{p}_{k-1}}.$$

To simplify this expression, in particular to get rid of the matrix-vector multiplications, multiply (6.12) from left with $\left(\underline{r}^{(k)}\right)^T$ such that one obtains, using $\underline{r}^{(k)} \perp \underline{r}^{(k-1)}$,

$$\left(\underline{r}^{(k)}\right)^T \underline{r}^{(k)} = 0 - \nu_k \left(\underline{r}^{(k)}\right)^T A \underline{p}_k \implies \nu_k = -\frac{\left(\underline{r}^{(k)}\right)^T \underline{r}^{(k)}}{\left(\underline{r}^{(k)}\right)^T A \underline{p}_k}.$$

This expression gives together with (6.16)

$$-\frac{\left(\underline{r}^{(k)}\right)^T A \underline{p}_k}{\underline{p}_k^T A \underline{p}_k} = \frac{\left(\underline{r}^{(k)}\right)^T \underline{r}^{(k)}}{\left(\underline{r}^{(k-1)}\right)^T \underline{r}^{(k-1)}},$$

such that

$$\mu_k = \frac{\left(\underline{r}^{(k-1)}\right)^T \underline{r}^{(k-1)}}{\left(\underline{r}^{(k-2)}\right)^T \underline{r}^{(k-2)}}, \quad k \geq 2. \quad (6.17)$$

The evaluation of this expression requires only two inner products but not any matrix-vector product. These considerations lead to Algorithm 6.8. \square

Algorithm 6.8. Conjugate Gradient (CG) method. Given a symmetric positive definite matrix $A \in \mathbb{R}^{n \times n}$, a right-hand side $\underline{b} \in \mathbb{R}^n$, an initial iterate $\underline{x}^{(0)} \in \mathbb{R}^n$, and a tolerance $\varepsilon > 0$.

1. $\underline{r}^{(0)} = \underline{b} - A \underline{x}^{(0)}$
2. $\underline{p}_1 = \underline{r}^{(0)}$
3. $k = 0$
4. **while** $\left\| \underline{r}^{(k)} \right\|_2 > \varepsilon$
5. $k = k + 1$
6. $\underline{s} = A \underline{p}_k$
7. $\nu_k = \frac{\left(\underline{r}^{(k-1)}\right)^T \underline{r}^{(k-1)}}{\underline{p}_k^T \underline{s}} \quad \% (6.16)$
8. $\underline{x}^{(k)} = \underline{x}^{(k-1)} + \nu_k \underline{p}_k \quad \% (6.13)$
9. $\underline{r}^{(k)} = \underline{r}^{(k-1)} - \nu_k \underline{s} \quad \% (6.12)$