March 26, 2018

Lecture 7 – Krylov methods for Hermitian matrices

Let $\mathbf{A}$ be an Hermitian non-singular $n \times n$ matrix.
For a given n vector $\mathbf{b}$, we want to solve $\mathbf{Ax} = \mathbf{b}$ for $\mathbf{x}$.

Recall that a complex positive definite (PD) matrix is Hermitian (see Exercise 0.29(a)).

A The Conjugate Gradient method

A pseudo code for a preconditioned version of CG (conjugate gradient method) is displayed in Alg. 7.1. Without preconditioner (take $\mathbf{M} = \mathbf{I}$ in Alg. 7.1), we have the standard CG algorithm. This method has very favourable properties: only three additional vectors have to be stored;[1] each step requires, next to the matrix-vector multiplication, only three vector updates and two inner products; when $\mathbf{A}$ is positive definite, then it computes an approximate solution in step k that is the best in the kth order Krylov subspace $\mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$, when measuring the error in the $\mathbf{A}$-norm; and in case of positive definiteness, the method is robust (the scalars for none of the denominators in the algorithm will be zero).

Exercise 7.1. **CG and breakdowns.**

(a) Show that CG can not break down if $\mathbf{A}$ is complex positive definite (PD), unless $\mathbf{r}_k = \mathbf{0}$ (which is called a **lucky breakdown**).

(b) In which step can CG break down if $\mathbf{A}$ is not PD? Why?

Let $\mathbf{A}$ be an $n \times n$ Hermitian matrix. Let $\mathbf{M}$ be a PD preconditioner. To preserve symmetry, the decomposition $\mathbf{M} = \mathbf{LL}^*$ could be used (cf., Exercise 0.29(c)) to precondition the problem $\mathbf{Ax} = \mathbf{b}$ as

$$(\mathbf{L}^{-1}\mathbf{AL}^{-*})\mathbf{y} = \mathbf{L}^{-1}\mathbf{b} \quad \text{and} \quad \mathbf{x} = \mathbf{L}^{-*}\mathbf{y}. \tag{7.1}$$

Here, $\mathbf{L}^{-*} \equiv (\mathbf{L}^*)^{-1}$. However, efficiency in solving the preconditioning system $\mathbf{Ms} = \mathbf{r}$ for $\mathbf{s}$ may require the call of a subroutine. For instance, $\mathbf{M}$ may represent the Laplace equation and solving is efficient with a multigrid method, fast Poisson solver or another iterative method. In these cases, a decomposition of $\mathbf{M}$ is not available. The following exercise (Exercise 7.2) explains how to incorporate a PD preconditioner in CG in case $\mathbf{A}$ is PD. The approach exploits $\mathbf{M}$, but is (mathematically) equivalent to applying standard CG to (7.1) (see Exercise 7.2(f). It relies on the fact that $\mathbf{M}^{-1}\mathbf{A}$ is self adjoint with respect to the $\mathbf{M}$-inner product (see Exercise 7.2(a)).

Exercise 7.2. Let $\mathbf{A}$ be Hermitian and let $\mathbf{M}$ be a PD preconditioner.

(a) Show that $\mathbf{M}^{-1}\mathbf{A}$ is self-adjoint in the $\mathbf{M}$-inner product, $(\mathbf{x}, \mathbf{y})_M \equiv \mathbf{y}^*\mathbf{Mx} \quad (\mathbf{x}, \mathbf{y} \in \mathbb{C}^n)$.

Now, assume $\mathbf{A}$ is PD.

(b) Prove that $\mathbf{M}^{-1}\mathbf{A}$ is PD in the $\mathbf{M}$-inner product.

(c) The CG method can be applied to the preconditioned system $\mathbf{M}^{-1}\mathbf{Ax} = \mathbf{M}^{-1}\mathbf{b}$ by working with the $\mathbf{M}$-inner product. Let $\mathbf{s}_k \equiv \mathbf{M}^{-1}\mathbf{r}_k$. Show that this leads to the following algorithm (ignoring the initialisation step):

$$\begin{aligned}
\mathbf{s}_k &= \mathbf{M}^{-1}\mathbf{r}_k \\
\rho_k &= (\mathbf{s}_k, \mathbf{s}_k)_M, \quad \beta_k = -\tfrac{\rho_k}{\rho_{k-1}} \\
\mathbf{u}_k &= \mathbf{s}_k - \beta_k \mathbf{u}_{k-1}, \quad \mathbf{c}_k = \mathbf{A}\mathbf{u}_k \\
\sigma_k &= (\mathbf{M}^{-1}\mathbf{c}_k, \mathbf{u}_k)_M, \quad \alpha_k = \tfrac{\rho_k}{\sigma_k} \\
\mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k \mathbf{u}_k, \quad \mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{c}_k
\end{aligned}$$

[1] The vectors $\mathbf{b}$ and $\mathbf{x}$ have to be stored anyway: they are part of the linear system. Note that $\mathbf{r}$ could take the place of $\mathbf{b}$ of $\mathbf{b}$ is not needed for other purposes (as for checking the accuracy $\|\mathbf{b} - \mathbf{Ax}_k\|_2$), which would reduce the number of additional vectors tot two.

PCG
Select $\mathbf{x}_0 \in \mathbb{C}^n$
$\mathbf{x} = \mathbf{x}_0$, $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$
$\mathbf{u} = \mathbf{0}$, $\rho = 1$
while $\|\mathbf{r}\|_2 > tol$ do
  Solve $\mathbf{M}\mathbf{c} = \mathbf{r}$ for $\mathbf{c}$
  $\rho' = \rho$, $\rho = \mathbf{c}^*\mathbf{r}$, $\beta = -\frac{\rho}{\rho'}$
  $\mathbf{u} \leftarrow \mathbf{c} - \beta\mathbf{u}$, $\mathbf{c} = \mathbf{A}\mathbf{u}$
  $\sigma = \mathbf{u}^*\mathbf{c}$, $\alpha = \frac{\rho}{\sigma}$
  $\mathbf{x} \leftarrow \mathbf{x} + \alpha\mathbf{u}$
  $\mathbf{r} \leftarrow \mathbf{r} - \alpha\mathbf{c}$
end while

ALGORITHM 7.1. PCG, preconditioned CG, i.e., CG with implicit preconditioning for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$ with residual accuracy *tol* using a preconditioner $\mathbf{M}$.
Both the system matrix $\mathbf{A}$ and the preconditioner $\mathbf{M}$ are assumed to be Hermitian, see Exercise 7.3.

(d) Show that $(\mathbf{s}_k, \mathbf{s}_k)_M = (\mathbf{r}_k, \mathbf{s}_k)_2$ and that $(\mathbf{M}^{-1}\mathbf{A}\mathbf{u}_k, \mathbf{u}_k)_M = (\mathbf{A}\mathbf{u}_k, \mathbf{u}_k)_2$.

(e) Show that (for PD matrices $\mathbf{A}$ and $\mathbf{M}$) the above algorithm is equivalent to the algorithm ALG. 7.1 (which is expressed in the standard inner product).

(f) Let $\mathbf{y}_k$ be the approximate solution obtained by applying standard CG to solve the equation in (7.1), with $\mathbf{y}_0 = \mathbf{L}^*\mathbf{x}_0$. Prove that the $\mathbf{x}_k$ produced by the above algoithm and the $\mathbf{y}_k$ relate as $\mathbf{x}_k = \mathbf{L}^{-*}\mathbf{y}_k$.

CG has been designed for positive definite matrices. However, it can also be used for non-definite Hermitian problems. Usually, the method converges fast, but the convergence is erratic and it can even break-down (cf., Exercise 7.1(b)) due to the fact that the LU-decomposition of the Lanczos tridiagonal matrix may not exists (see Exercise 7.5). For non-definite matrices, CG does not minimise. Its good convergence properties are explained by the fact that CG puts its residuals orthogonal to a sequence of growing spaces, as we will learn in the next exercise, where we also incorporate a preconditioner $\mathbf{M}$ that is required to be Hermitian, but not PD.

Exercise 7.3. **CG for symmetric non-definite systems.** For square non-singular matrices $\mathbf{A}$ and $\mathbf{M}$, the CG recurrence relations are determined by the coupled two-term recurrences

$$\begin{aligned} \mathbf{u}_k &= \mathbf{M}^{-1}\mathbf{r}_k - \beta_k\,\mathbf{u}_{k-1} \\ \mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k\,\mathbf{A}\,\mathbf{u}_k \end{aligned} \tag{7.2}$$

(with $\mathbf{u}_{-1} \equiv \mathbf{0}$) and the orthogonality requirement

$$\mathbf{r}_k,\ \mathbf{A}\mathbf{u}_k \perp \mathbf{M}^{-1}\mathbf{r}_{k-1}. \tag{7.3}$$

Note that (7.3) describes orthogonality with respect to the $\mathbf{M}^{-1}$ inner product in case $\mathbf{M}$ is positive definite, see Exercise 7.2. However, here, we do *not* assume definiteness. We only assume that both $\mathbf{A}$ and $\mathbf{M}$ are square, non-singular, Hermitian.

(a) Show that

$$\text{span}(\mathbf{u}_0, \ldots, \mathbf{u}_{k-1}) = \mathbf{M}^{-1}\mathcal{K}_k(\mathbf{A}\mathbf{M}^{-1}, \mathbf{r}_0) \quad \text{and} \quad \text{span}(\mathbf{r}_0, \ldots, \mathbf{r}_k) = \mathcal{K}_{k+1}(\mathbf{A}\mathbf{M}^{-1}, \mathbf{r}_0).$$

(b) Prove that

$$\mathbf{r}_k,\ \mathbf{A}\mathbf{u}_k \perp \mathbf{M}^{-1}\mathcal{K}_k(\mathbf{A}\mathbf{M}^{-1}, \mathbf{r}_0). \tag{7.4}$$

(c) Put

$$\rho_k \equiv \mathbf{r}_k^* \mathbf{M}^{-1} \mathbf{r}_k \quad \text{and} \quad \sigma_k = \mathbf{u}_k^* \mathbf{A} \mathbf{u}_k.$$

Show that

$$\alpha_k = \frac{\rho_k}{\sigma_k} \quad \text{and} \quad \beta_k = -\frac{\rho_k}{\rho_{k-1}}.$$

Conclusion. *It suffices to put* $\mathbf{r}_k$ *and* $\mathbf{A}\mathbf{u}_k$ *orthogonal to only one vector* (namely, $\mathbf{M}^{-1}\mathbf{r}_{k-1}$) *at the cost of only two inner products* (ρ_k and σ_k, ρ_{k-1} is available from the previous step), *to get a residual* $\mathbf{r}_k$ *that is orthogonal to a* k*-dimensional space* ($\mathbf{M}^{-1}\mathcal{K}_k(\mathbf{A}\mathbf{M}^{-1}, \mathbf{r}_0)$). *We can efficiently produce a sequence of residuals that are orthogonal to a sequence of 'growing' spaces.*

(d) Derive ALG. 7.1.

(e) Assume $\mathbf{A}$ is also positive definite. Then $\|\mathbf{y}\|_{A^{-1}} \equiv \sqrt{\mathbf{y}^*(\mathbf{A}^*)^{-1}\mathbf{y}}$ defines a norm (the A^{-1}-norm). Prove that that CG (as described here) finds the smallest residual in A^{-1}-norm: $\|\mathbf{r}_k\|_{A^{-1}} \leq \|\mathbf{b} - \mathbf{A}\widetilde{\mathbf{x}}\|_{A^{-1}}$ for all $\widetilde{\mathbf{x}} \in \mathbf{x}_0 + \mathbf{M}^{-1}\mathcal{K}_k(\mathbf{A}\mathbf{M}^{-1}, \mathbf{r}_0)$ and the smallest error in A-norm.

B Lanczos and CG

In case $\mathbf{A}$ is Hermitian, the $(k+1) \times k$ upper Hessenberg matrix $\underline{H}_k$ in the Arnoldi relation $\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\underline{H}_k$ can shown to be tri-diagonal. To emphasise tri-diagonality, $\underline{H}_k$ is denoted by $\underline{T}_k$ and the Arnoldi relation is referred to as the **Lanczos relation**:

$$\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\underline{T}_k \quad \text{with } \mathbf{V}_k \; n \times k \text{ orthonormal.} \tag{7.5}$$

$\underline{T}_k$ is the **Lanczos matrix**. Tri-diagonality of $\underline{T}_k$ leads to the **Lanczos recursion**

$$\mathbf{A}\mathbf{v}_k = \beta_k \mathbf{v}_{k-1} + \alpha_k \mathbf{v}_k + \beta_{k+1}\mathbf{v}_{k+1} \quad (\mathbf{v}_{k+1} \perp \mathbf{v}_k, \; \|\mathbf{v}_{k+1}\|_2 = 1), \tag{7.6}$$

with α_k the kth diagonal entry of $\underline{T}_k$ and β_k its co-diagonal coefficients. Note that the coefficient α_k orthogonalises $\mathbf{A}\mathbf{v}_k$ against $\mathbf{v}_k$, while β_{k+1} leads to a normalised vector $\mathbf{v}_{k+1}$, and the normalisation coefficient β_k for $\mathbf{v}_k$ implicitly orthogonalises $\mathbf{A}\mathbf{v}_k$ against $\mathbf{v}_{k-1}$: as indicated in (7.6), $\mathbf{v}_{k+1} \perp \mathbf{v}_k, \mathbf{v}_{k-1}$, $\|\mathbf{v}_{k+1}\|_2 = 1$. The algorithm that explicitly uses (7.6) to determine the Lanczos relation is called the **Lanczos algorithm**; see ALG. 7.2. The vectors $\mathbf{v}_k$ when computed with the Lanczos algorithm are called **Lanczos vectors**.

Exercise 7.4. **The Lanczos relation.** Suppose $\mathbf{A}$ is Hermitian (i.e., $\mathbf{A}^* = \mathbf{A}$).

For a normalised vector $\mathbf{v}_1$, consider the Arnoldi relation

$$\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\underline{H}_k,$$

where $\mathbf{V}_{k+1}$ is an orthonormal $n \times k$ matrix and $\underline{H}_k$ is a $(k+1) \times k$ Hessenberg.

(a) Prove that $\underline{H}_k$ is tri-diagonal. Put $\underline{T}_k \equiv \underline{H}_k$.

(b) Show that the Arnoldi vectors satisfy the three term recurrence relation (7.6). Conversely, the orthonormality condition in (7.6) leads to Arnoldi vectors.

(c) Derive the Lanczos algorithm ALG. 7.2 by making use of the fact that the Lanczos vectors form an orthonormal basis of the Krylov subspace.

The following exercise explains how the Lanczos relation (7.5) can be obtained as a side product of the CG process. It also explains how CG can be viewed as an efficient implementation of Lanczos combined with a method for solving the projected (lower dimensional) system $\underline{T}_k y = \|\mathbf{r}_0\|_2 e_1$ with an LU-decomposition (Gaussian elimination).

Exercise 7.5. **From CG to Lanczos.** Consider the two coupled two term CG-recurrences

$$\begin{aligned} \mathbf{u}_k &= \mathbf{r}_k - \beta_k \mathbf{u}_{k-1} \qquad (\mathbf{A}\mathbf{u}_k \perp \mathbf{r}_{k-1}) \\ \mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{u}_k \quad (\mathbf{r}_{k+1} \perp \mathbf{r}_k) \end{aligned} \tag{7.7}$$

LANCZOS

$\rho = \|\mathbf{r}_0\|_2, \quad \mathbf{v}_1 = \mathbf{r}_0/\rho$

$\beta_1 = 0, \quad \mathbf{v}_0 = \mathbf{0}$

for $k = 1, 2, \ldots$ do

$\widetilde{\mathbf{v}} = \mathbf{A}\,\mathbf{v}_k - \beta_k\,\mathbf{v}_{k-1}$

$\alpha_k = \mathbf{v}_k^*\,\widetilde{\mathbf{v}}, \quad \widetilde{\mathbf{v}} \leftarrow \widetilde{\mathbf{v}} - \alpha_k\,\mathbf{v}_k$

$\beta_{k+1} = \|\widetilde{\mathbf{v}}\|_2, \quad \mathbf{v}_{k+1} = \widetilde{\mathbf{v}}/\beta_{k+1}$

end while

ALGORITHM 7.2. The Lanczos algorithm [Lanczos '50] for computing the Lanczos decomposition $\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\,\underline{T}_k$ for a Hermitian matrix $\mathbf{A}$. The matrix $\mathbf{V}_k = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$ is orthonormal and $\underline{T}_k$ is $(k+1) \times k$ tridiagonal with diagonal entry (i,i) equal to α_i and both codiagonal entries $(i, j+1)$ and $(i+1, j)$ equal to β_{j+1}.

The scalars β_k and α_k are determined by the orthogonality restrictions as indicated in (7.7): $\alpha_k = \frac{\rho_k}{\sigma_k}$ and $\beta_k = -\frac{\rho_k}{\rho_{k-1}}$, where $\rho_k = \mathbf{r}_k^*\mathbf{r}_k = \|\mathbf{r}_k\|_2^2$ and $\sigma_k \equiv \mathbf{u}_k^*\mathbf{A}\mathbf{u}_k$.

(a) Prove that the $\mathbf{r}_j$ $(j < k)$ form an orthogonal basis of $\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$, while the $\mathbf{u}_j$ $(j < k)$ form an $\mathbf{A}$-orthogonal basis. In particular, the $\mathbf{v}_j \equiv \frac{1}{\sqrt{\rho_j}}\mathbf{r}_j$ form an orthonormal basis, while $\frac{1}{\sqrt{\sigma_j}}\mathbf{u}_j$ form an $\mathbf{A}$ orthonormal basis.

(b) Show that $\mathbf{A}\mathbf{u}_k = \frac{1}{\alpha_k}(\mathbf{r}_k - \mathbf{r}_{k+1})$ and

$$\mathbf{A}\mathbf{r}_k = \frac{1}{\alpha_k}(\mathbf{r}_k - \mathbf{r}_{k+1}) + \frac{\beta_k}{\alpha_{k-1}}(\mathbf{r}_{k-1} - \mathbf{r}_k). \tag{7.8}$$

With $\mathbf{v}_k \equiv \frac{1}{\sqrt{\rho_k}}\mathbf{r}_k$, show that the Lanczos relation holds

$$\mathbf{A}\mathbf{v}_k = \gamma_{k+1}\mathbf{v}_{k+1} + \alpha'_k\mathbf{v}_k + \gamma_k\mathbf{v}_{k-1}, \quad \text{where} \quad \gamma_{k+1} = -\frac{\sqrt{\rho_{k+1}}}{\sqrt{\rho_k}\,\alpha_k},\ \alpha'_k = \frac{1}{\alpha_k} - \frac{\beta_k}{\alpha_{k-1}}, \tag{7.9}$$

(c) Put $\mathbf{R}_k \equiv [\mathbf{r}_0, \ldots \mathbf{r}_{k-1}]$ and $\mathbf{U}_k \equiv [\mathbf{u}_0, \ldots \mathbf{u}_{k-1}]$. Let $\underline{J}_k$ be the $(k+1)\times k$ bi-diagonal matrix with 1 on the main diagonal and -1 on the first lower diagonal, let B_k be the $k \times k$ bi-diagonal matrix with 1 on the main diagonal and β_k on the first upper diagonal. Show that (7.8) reads as

$$\mathbf{R}_k = \mathbf{U}_k B_k, \quad \mathbf{A}\mathbf{U}_k D_\alpha = \mathbf{R}_{k+1}\underline{J}_k. \tag{7.10}$$

Here, $D_\alpha \equiv \mathrm{diag}(\alpha_0, \ldots, \alpha_{k-1})$. Conclude that $\mathbf{A}\mathbf{R}_k = \mathbf{R}_{k+1}\underline{J}_k D_\alpha^{-1} B_k$ represents (7.8). Also, prove that $\mathbf{A}\mathbf{U}_k = \mathbf{U}_{k+1}B_{k+1}\underline{J}_k D_\alpha^{-1}$.
Let $D_\rho \equiv \mathrm{diag}(\sqrt{\rho_0}, \ldots, \sqrt{\rho_{k-1}})$. We suppress the index k for diagonal matrices. The dimension should be obvious from the context. Hence, (7.9) is represented by

$$\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\,\underline{T}_k \quad \text{with} \quad \underline{T}_k \equiv D_\rho\,\underline{J}_k D_\alpha^{-1} B_k D_\rho^{-1}.$$

Note that $\underline{T}_k$ is a product of a lower triangular $D_\rho\,\underline{J}_k D_\alpha^{-1}$ and an upper triangular matrix $B_k D_\rho^{-1}$. Is this a standard LU-decomposition? Note that the (any) LU-decomposition of $\underline{T}_k$ fails iff a $1/\alpha_j$ is 0, that is, if a $\sigma_j = 0$. For this reason, $\sigma_j = 0$ is referred to as a **breakdown of the LU-decomposition**.

(d) Show that $B_k = D_\rho^{-2} J_k^* D_\rho^2$, where J_k is the $k \times k$ upper block of $\underline{J}_k$. Give a Cholesky decomposition of $\underline{T}_k$.

(e) Suppose $\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\,\underline{T}_k$ and $\underline{T}_k = \underline{L}_k U_k$ is an LU-decomposition of $\underline{T}_k$. Show that CG computes $\mathbf{x}_k$ as $\mathbf{x}_k = (\mathbf{V}_k U_k^{-1})(L_k^{-1}[\sqrt{\rho_0}\,e_1]) = \mathbf{U}_k(L_k^{-1}[\sqrt{\rho_0}\,e_1])$. Note that the fact that CG does not rely on the standard LU-decomposition of the Lanczos tridiagonal matrix $\underline{T}_k$ only affects the scaling of the $\mathbf{u}_k$.

The Lanczos relation (7.5) can be used for iteratively solving linear systems. As we saw in Exercise 7.5, CG is an implementation of this idea that relies on an LU-decomposition of $\underline{T}_k$. It leads to **two coupled two term recurrences** for iterating residuals. Since the residuals are multiples of the Lanczos vectors (the columns of $\mathbf{V}_k$), CG can also be viewed as an implementation for computing the Lanczos relation with two coupled two term recurrences for generating the Lanczos vectors, cf., Exercise 7.5. In the following exercise, we will see a method, the so-called **Lanczos' method**, that relies on one **three term recurrence** relation for generating residuals.

Exercise 7.6. **Lanczos' method.** With $\mathbf{v}_{-1} \equiv \mathbf{0}$ and $\mathbf{v}_0 \equiv \mathbf{b}/\|\mathbf{b}\|_2$, consider the Lanczos relation

$$\mathbf{A}\mathbf{v}_k = \beta_{k+1}\mathbf{v}_{k+1} + \alpha_k\mathbf{v}_k + \beta_k\mathbf{v}_{k-1} \quad (\mathbf{v}_{k+1} \perp \mathbf{v}_k, \mathbf{v}_{k-1}) \qquad (k = 0, 1, 2, \ldots)$$

with scalars β_k, β_{k+1} and α_k such that the orthogonality restrictions as indicated above hold. Then, the vectors $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{k+1}$ form an orthonormal system,

Let (ρ_k) be sequence in $(0, \infty)$. Put $\mathbf{r}_k \equiv \rho_k\mathbf{v}_k$.

(a) Show that

$$\gamma_k\mathbf{r}_{k+1} = \mathbf{A}\mathbf{r}_k - \alpha_k\mathbf{r}_k - \beta'_k\mathbf{r}_{k-1} \quad \text{for} \quad \gamma_k \equiv \beta_{k+1}\frac{\rho_k}{\rho_{k+1}}, \quad \beta'_k \equiv \beta_k\frac{\rho_k}{\rho_{k-1}}$$

Show that for each k there is a polynomial p_k of degree k such that $\mathbf{r}_k = p_k(\mathbf{A})\mathbf{b}$.
In particular, $\zeta p_k(\zeta) = \gamma_k p_{k+1}(\zeta) + \alpha_k p_k(\zeta) + \beta'_k p_{k-1}(\zeta)$ for all k $(\zeta \in \mathbb{C})$.

(b) Show that $\quad p_k(0) = 1$ for all $k \quad \Leftrightarrow \quad \gamma_k + \alpha_k + \beta'_k = 0$ for all k.
Show that
$p_k(0) = 1 \quad \Leftrightarrow \quad \mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$ for some polynomial q_k of degree $< k$, where $\mathbf{x}_k = q_k(\mathbf{A})\mathbf{b}$.
Hence, $\mathbf{r}_k$ are residuals (all k) $\quad \Leftrightarrow \quad \gamma_k + \alpha_k + \beta'_k = 0$ (all k).
Note that we here have a special case of (5.8). See also Exercise 5.25.

(c) Assume that $\gamma_k + \alpha_k + \beta'_k = 0$ all k. Note that $\gamma_k\mathbf{b} + \alpha_k\mathbf{b} + \beta'_k\mathbf{b} = \mathbf{0}$. Prove that

$$\begin{aligned}
&\mathbf{r}_0 = \mathbf{b}, \quad \mathbf{x}_0 = \mathbf{0} \\
&\gamma_0\mathbf{r}_1 = \mathbf{A}\mathbf{r}_0 - \alpha_0\mathbf{r}_0 \perp \mathbf{r}_0, \quad \gamma_0 + \alpha_0 = 0, \quad \gamma_0\mathbf{x}_1 = -\mathbf{r}_0 \\
&\gamma_k\mathbf{r}_{k+1} = \mathbf{A}\mathbf{r}_k - \alpha_k\mathbf{r}_k - \beta'_k\mathbf{r}_{k-1} \perp \mathbf{r}_k, \mathbf{r}_{k+1}, \\
&\gamma_k + \alpha_k + \beta'_k = 0, \quad \gamma_k\mathbf{x}_{k+1} = -\mathbf{r}_k - \alpha_k\mathbf{x}_k - \beta'_k\mathbf{x}_{k-1}
\end{aligned} \tag{7.11}$$

(d) Show that the condition $\widetilde{\mathbf{r}}_{k+1} \equiv \mathbf{A}\mathbf{r}_k - \alpha_k\mathbf{r}_k - \beta'_k\mathbf{r}_{k-1} \perp \mathbf{r}_k, \mathbf{r}_{k-1}$ determines α_k and β'_k, while $\gamma_k \equiv -\alpha_k - \beta'_k$ and $\mathbf{r}_{k+1} \equiv \widetilde{\mathbf{r}}_{k+1}/\gamma_k$, defines $\mathbf{r}_{k+1}$.

(e) Of course, there is no need to compute the $\mathbf{r}_k$ explicitly. Prove that (7.11) is equivalent to

$$\begin{aligned}
&\mathbf{v}_{-1} = \mathbf{0}, \quad \mathbf{v}_0 = \mathbf{b}/\|\mathbf{b}\|_2, \quad \mathbf{x}_{-1} = \mathbf{0} \quad \mathbf{x}_0 = \mathbf{0} \\
&\mathbf{A}\mathbf{v}_k = \beta_{k+1}\mathbf{v}_{k+1} + \alpha_k\mathbf{v}_k + \beta_k\mathbf{v}_{k-1} \quad (\mathbf{v}_{k+1} \perp \mathbf{v}_k, \mathbf{v}_{k-1}) \\
&\frac{\beta_{k+1}}{\rho_{k+1}} = -\frac{\alpha_k}{\rho_k} - \frac{\beta_k}{\rho_{k-1}}, \quad \frac{\beta_{k+1}}{\rho_{k+1}}\mathbf{x}_{k+1} = -\mathbf{v}_k - \frac{\alpha_k}{\rho_k}\mathbf{x}_k - \frac{\beta_k}{\rho_{k-1}}\mathbf{x}_{k-1}
\end{aligned}$$

(f) Derive an algorithm for an iterative solver based on the above ideas (**Lanczos' method**).

C MINRES and SYMMLQ for indefinite systems

CG can be viewed as being a method based on Lanczos recursions (see Exercise 7.5), whereas MINRES and SYMMLQ are methods that are explicitly built on top of the Lanczos recursion. These methods avoid the breakdown dangers of CG and they minimise the norm of residuals in case of MINRES and the norm of errors in case of SYMMLQ. Both methods converge 'smoothly', where the convergence of CG can be erratic for Hermitian indefinite systems.

MINRES versus GMRES. If $\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\,\underline{H}_k$ is the kth Arnoldi relation and $\mathbf{x}_0 = \mathbf{0}$, then GMRES computes the kth approximate solution $\mathbf{x}_k$ as

$$\mathbf{x}_k = \|\mathbf{b}\|_2\,\mathbf{V}_k(R_k^{-1}(\underline{Q}_k{}^* e_1)),$$

where $\underline{H}_k = \underline{Q}_k R_k$ is the economical QR-decomposition of $\underline{H}_k$. In case $\mathbf{A}$ is Hermitian, $\underline{H}_k$ is tri-diagonal (see Exercise 7.7Exercise 7.7). This allows us to compute the $(\mathbf{v}_k)$ with a short recurrence: Lanczos is an efficient variant of Arnoldi for Hermitian systems. To be able to use short recurrences also for the numerical solution of linear systems, MINRES computes $\mathbf{x}_k$ as

$$\mathbf{x}_k = \|\mathbf{b}\|_2\,(\mathbf{V}_k R_k^{-1})(\underline{Q}_k{}^* e_1),$$

see Exercise 7.8: the only difference between GMRES and MINRES is the order of the multiplication of the three matrices $\mathbf{V}_k$, R_k^{-1} and $\underline{Q}_k{}^*$. Since R_k is upper triangular and tridiagonal, the columns $\mathbf{w}_j$ of the matrix $\mathbf{W}_k \equiv \mathbf{V}_k R_k^{-1}$ can be computed from $\mathbf{V}_k$ by a three-term recurrence relation (use the fact that $\mathbf{W}_k R_k = \mathbf{V}_k$). Moreover, to compute $\mathbf{w}_k$ only the vectors $\mathbf{v}_k$, $\mathbf{w}_{k-1}$ and $\mathbf{w}_{k-2}$ are needed plus the last colum of R_k. From a mathematical point of view, GMRES and MINRES are the same. From a computational point of view, they are very different: MINRES uses three-term recurrences and only stores seven n-vectors, while in GMRES the recurrences and the storage requirements are proportional to the step number. On the other hand, MINRES is slightly less stable than GMRES.

Exercise 7.7. **Lanczos and decompositions.**
For a normalised vector $\mathbf{v}_1$, consider the Lanczos relation (7.5).

(a) Show that the LU-decomposition $\underline{T}_k = \underline{L}_k\,U_k$ exists if $\mathbf{A}$ is positive definite. Show that without this additional restriction, the LU-decomposition need not exist. (Hint: $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$).

(b) Prove that both $\underline{L}_k$ and U_k are bi-diagonal matrices (if the LU-decomposition of $\underline{T}_k$ exists). If the LU-decomposition does not exists at step k, then we refer to this situation as the **breakdown of the LU-decomposition**.

(c) Show that the QR-decomposition $\underline{T}_k = Q_{k+1}\underline{I}_k R_k$ exists. Here, R_k is upper triangular, Q_{k+1} is unitary and $\underline{I}_k$ is the $(k+1)\times k$ identity matrix with a last row of zeros.

(d) Prove that R_k is tri-diagonal. Let $g(\phi)$ be the *Givens rotation*

$$g(\phi) \equiv \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{bmatrix}$$

Show that Q_{k+1} can be obtained as $Q_{k+1} = G_1 \cdot \ldots \cdot G_k$ with G_j the $k+1$ by $k+1$ identity matrix, except for the 2×2 block at the entries (p,q) with $p,q \in \{j, j+1\}$, which is a Givens rotation $g(\phi_j)$.

(e) We will now show that R_{k+1} and the decomposition $Q_{k+2} = G_1 \cdot \ldots \cdot G_k G_{k+1}$ can be obtained as simple updates of R_k and of the $G_1, \ldots, G_k$ from Q_{k+1}. First, note that $\underline{T}_{k+1}$ "equals" $\underline{T}_k$ plus a new column t_{k+1}. Show that $G_k^* \cdot \ldots \cdot G_1^* t_{k+1} = G_k^* G_{k-1}^* t_{k+1}$ (here we assume the G_j to be extended to match dimensions). Give an expression for G_{k+1} in terms of the coordinates of $t'_{k+1} \equiv G_k^* G_{k-1}^* t_{k+1}$. Give an expression for the last column of R_{k+1}.

(f) Discuss the situation where $r_{k,k} \equiv e_k^* R_k e_k$ is zero.

In Exercise 7.8 and Exercise 7.9 below, we follow the notation and we use the results of Exercise 7.7. For $\mathbf{x}_0$, let $\mathbf{r}_0 \equiv \mathbf{b} - \mathbf{A}\mathbf{x}_0$. The Arnoldi relation is generated by $\mathbf{v}_1 \equiv \mathbf{r}_0/\|\mathbf{r}_0\|_2$. For $\mathbf{x}_k$, $\mathbf{r}_k \equiv \mathbf{b} - \mathbf{A}\mathbf{x}_k$.

Exercise 7.8. **MINRES.** For a k-vector y_k, let $\mathbf{x}_k \equiv \mathbf{x}_0 + \mathbf{V}_k\,y_k$. Then $\mathbf{r}_k = \mathbf{r}_0 - \mathbf{A}\mathbf{V}_k\,y_k$.
MINRES is characterised by the condition

$$y_k = \mathrm{argmin}_y \|\mathbf{r}_0 - \mathbf{A}\mathbf{V}_k\,y\|_2.$$

MINRES

$\mathbf{x} = \mathbf{0}, \quad \mathbf{r} = \mathbf{b}, \quad \rho = \|\mathbf{r}\|_2, \quad \mathbf{v} = \mathbf{r}/\rho$

$\beta = 0, \quad \widetilde{\beta} = 0, \quad c = -1, \quad s = 0$

$\mathbf{v}_{\mathtt{old}} = \mathbf{0}, \quad \mathbf{w} = \mathbf{0}, \quad \widetilde{\widetilde{\mathbf{w}}} = \mathbf{v}$

`while` $|\rho| > tol$ `do`

%% *Lanczos*

$\widehat{\mathbf{v}} = \mathbf{A}\mathbf{v} - \beta\,\mathbf{v}_{\mathtt{old}}, \quad \alpha = \mathbf{v}^*\,\widehat{\mathbf{v}}, \quad \widehat{\mathbf{v}} \leftarrow \widehat{\mathbf{v}} - \alpha\,\mathbf{v}$

$\beta = \|\widehat{\mathbf{v}}\|_2, \quad \mathbf{v}_{\mathtt{old}} = \mathbf{v}, \quad \mathbf{v} = \widehat{\mathbf{v}}/\beta$

%% *QR-decomposition of the Lanczos matrix*

$l_1 = s\,\alpha - c\,\widetilde{\beta}, \quad l_2 = s\,\beta, \quad \widetilde{\alpha} = -s\,\widetilde{\beta} - c\,\alpha, \quad \widetilde{\beta} = c\,\beta$

$l_0 = \sqrt{\widetilde{\alpha}^2 + \beta^2}, \quad c = \widetilde{\alpha}/l_0, \quad s = \beta/l_0$

%% *The search vector*

$\widetilde{\mathbf{w}} = \widetilde{\widetilde{\mathbf{w}}} - l_1\,\mathbf{w}, \quad \widetilde{\widetilde{\mathbf{w}}} = \mathbf{v} - l_2\,\mathbf{w}, \quad \mathbf{w} = \widetilde{\mathbf{w}}/l_0$

%% *The approximate solution*

$\mathbf{x} \leftarrow \mathbf{x} + (\rho\,c)\,\mathbf{w},$

%% *The residual norm*

$\rho \leftarrow s\,\rho$

`end while`

ALGORITHM 7.3. MINRES [Paige & Saunders '75] for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$ with residual accuracy *tol*. $\mathbf{A}$ is an Hermitian matrix.

(a) Show that

$$\mathbf{x}_k = \mathbf{x}_0 + \|\mathbf{r}_0\|_2 \left(\mathbf{V}_k\,R_k^{-1}\right)\left(\underline{I}_k{}^* Q_{k+1}^* e_1\right).$$

(b) Put $\mathbf{W}_k \equiv \mathbf{V}_k\,R_k^{-1}$. Show that the relation $\mathbf{W}_k\,R_k = \mathbf{V}_k$ can be used to compute the column vectors $\mathbf{w}_j$ of $\mathbf{W}_k$ with a three term vector recursion:

$$\mathbf{w}_k = \frac{1}{r_{k,k}}(\mathbf{v}_k - r_{k-1,k}\mathbf{w}_{k-1} - r_{k-2,k}\mathbf{w}_{k-2}).$$

(c) Put $z_{k+1} \equiv \|\mathbf{r}_0\|_2\,Q_{k+1}^* e_1$. Let z'_{k+1} be the vector of the first k coordinates of z_{k+1} and let ζ_{k+1} be the last coordinate: $z_{k+1} = ((z'_{k+1})^{\mathrm{T}}, \zeta_{k+1})^{\mathrm{T}}$. Show that

$$z_{k+1} = \begin{bmatrix} z'_k \\ \cos(\phi_k)\zeta_k \\ \sin(\phi_k)\zeta_k \end{bmatrix}, \qquad z'_{k+1} = \begin{bmatrix} z'_k \\ \cos(\phi_k)\zeta_k \end{bmatrix}, \qquad \zeta_{k+1} = \sin(\phi_k)\zeta_k.$$

(d) Prove that

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \cos(\phi_k)\zeta_k\mathbf{w}_k.$$

(e) Derive algorithm ALG. 7.3 for MINRES.

(f) prove that

$$\|\mathbf{r}_k\|_2 = \|\mathbf{r}_0\|_2\,\|Q_{k+1}^* e_1 - \underline{I}_k Q_{k+1}^* e_1\| = |\zeta_{k+1}|.$$

(g) Explain the naming 'MINRES' of the method.

In contrast to MINRES, the $\mathbf{x}_k$ in SYMMLQ is not taken from $\mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$, but from $\mathbf{x}_0 + \mathbf{A}\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$. This strategy allows SYMMLQ to find an approximate solution in the search

SYMMLQ

$\mathbf{x} = \mathbf{0}$, $\mathbf{r} = \mathbf{b}$, $\rho = \|\mathbf{r}\|_2$, $\mathbf{v} = \mathbf{r}/\rho$

$\beta = 0$, $\widetilde{\beta} = 0$, $c = -1$, $s = 0$, $\kappa = \rho$

$\mathbf{v}_{\mathtt{old}} = \mathbf{0}$, $\mathbf{w} = \mathbf{v}$, $g = 0$, $\widetilde{\widetilde{g}} = \rho$

`while` $\kappa > tol$ `do`

%% *Lanczos*

$\widehat{\mathbf{v}} = \mathbf{A}\,\mathbf{v} - \beta\,\mathbf{v}_{\mathtt{old}}$, $\alpha = \mathbf{v}^*\,\widehat{\mathbf{v}}$, $\widehat{\mathbf{v}} \leftarrow \widehat{\mathbf{v}} - \alpha\,\mathbf{v}$

$\beta = \|\widehat{\mathbf{v}}\|_2$, $\mathbf{v}_{\mathtt{old}} = \mathbf{v}$, $\mathbf{v} = \widehat{\mathbf{v}}/\beta$

%% *QR-decomposition of the Lanczos matrix*

$l_1 = s\,\alpha - c\,\widetilde{\beta}$, $l_2 = s\,\beta$, $\widetilde{\alpha} = -s\,\widetilde{\beta} - c\,\alpha$, $\widetilde{\beta} = c\,\beta$

$l_0 = \sqrt{\widetilde{\alpha}^2 + \beta^2}$, $c = \widetilde{\alpha}/l_0$, $s = \beta/l_0$

%%

$\widetilde{g} = \widetilde{\widetilde{g}} - l_1\,g$, $\widetilde{\widetilde{g}} = -l_2\,g$, $g = \widetilde{g}/l_0$

%% *The approximate solution*

$\mathbf{x} \leftarrow \mathbf{x} + (g\,c)\,\mathbf{w} + (g\,s)\,\mathbf{v}$

%% *The search vector*

$\mathbf{w} \leftarrow s\,\mathbf{w} - c\,\mathbf{v}$,

%% *The residual norm*

$\kappa = \sqrt{\widetilde{g}^2 + \widetilde{\widetilde{g}}^2}$

`end while`

ALGORITHM 7.4. SYMMLQ [Paige & Saunders '75] for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$ with residual accuracy *tol*. $\mathbf{A}$ is an Hermitian matrix. Note that the "Lanczos" part and the update of the QR-decomposition of the Lanczos matrix is the same as for MINRES.

subspace that minimises the norm of the error rather than the norm of the residual (as in MINRES).

Exercise 7.9. **SYMMLQ.** For a k-vector y'_k, let $\mathbf{x}_k \equiv \mathbf{x}_0 + \mathbf{A}\mathbf{V}_k\,y'_k$.
SYMMLQ minimises the norm of the error. Hence,

$$y'_k = \mathrm{argmin}_{y'}\|\mathbf{x} - \mathbf{x}_0 - \mathbf{A}\mathbf{V}_k\,y'\|_2.$$

Note that, with $y_k \equiv \underline{T}_k\,y'_k$, we have that

$$\mathbf{x}_k \equiv \mathbf{x}_0 + \mathbf{A}\mathbf{V}_k\,y'_k = \mathbf{x}_0 + \mathbf{V}_{k+1}\,\underline{T}_k\,y'_k = \mathbf{x}_0 + \mathbf{V}_{k+1}\,y_k.$$

(a) Show that y'_k satisfies $\mathbf{x} - \mathbf{x}_0 - \mathbf{V}_{k+1}\underline{T}_k y'_k \perp \mathbf{A}\mathbf{V}_k$, whence $\|\mathbf{r}_0\|_2\,e_1 - \underline{T}_k{}^*\underline{T}_k y'_k = 0$. Hence,

$$y_k = \|\mathbf{r}_0\|_2\,\underline{T}_k(\underline{T}_k{}^*\underline{T}_k)^{-1}e_1, \quad \text{and} \quad \mathbf{x}_k = \mathbf{x}_0 + (\mathbf{V}_{k+1}Q_{k+1})\,(\underline{I}_k\,z_k)\,,$$

where z_k solves $R_k^* z_k = \|\mathbf{r}_0\|_2\,e_1$.
(b) Put $\mathbf{W}_{k+1} \equiv \mathbf{V}_{k+1}Q_{k+1}$. Show that $Q_{k+1}^* e_{k+1} = G_k^* e_{k+1}$. Use $\mathbf{W}_{k+1}Q_{k+1}^* = \mathbf{V}_{k+1}$ to show that the columns of $\mathbf{W}_{k+1}$ satisfy a two term vector recursion:

$$\mathbf{w}_{k+1} = \frac{1}{\cos(\phi_k)}(\mathbf{v}_{k+1} + \sin(\phi_k)\mathbf{w}_k).$$

(c) With z'_k the vector of the first $k-2$ coordinates of z_k and ζ'_k, ζ_k the last two coordinates, $z_k = ((z'_k)^{\mathrm{T}}, \zeta'_k, \zeta_k)^{\mathrm{T}}$, show that $\zeta'_{k+1} = \zeta_k$ and express ζ_{k+1} in terms of the quantities from z_k and the last column of R_{k+1} (this a three term scalar recurrence relation).

(d) Show that $\mathbf{x}_k = \mathbf{x}_{k-1} + \zeta_k \, \mathbf{w}_k$.

(e) Derive algorithm ALG. 7.4 for SYMMLQ.

(f) For a stopping criterion, we need $\|\mathbf{r}_k\|_2$. Show that

$$\|\mathbf{r}_k\|_2 = \|\mathbf{r}_0\|_2 \, \|e_1 - \underline{T}_{k+1}\underline{T}_k(\underline{T}_k{}^*\underline{T}_k)^{-1}e_1\|_2.$$

Show that for some $k+1$ vectors t_k and s_k we have that $\underline{T}^*_{k+1} = [\underline{T}_k, t_k, s_k]$. Prove that

$$\|\mathbf{r}_k\|_2 = \sqrt{|t_k^* y_k|^2 + |s_k^* y_k|^2}.$$

Give efficient formulae to compute $|t_k^* y_k|$ and $|s_k^* y_k|$.

(g) Note that y'_k is the minimal-norm solution of the underdetermined system $\underline{T}_k{}^* y' = \|\mathbf{r}_0\|_2 e_1$. The QR decomposition can be used to compute the least-square solution (minimal residual solution) of overdetermined systems. Similarly, the LQ-decomposition can be used to compute the minimal-norm solution of underdetermined systems. Explain de naming 'SYMMLQ' of the method.

Convergence of SYMMLQ versus MINRES. SYMMLQ finds an approximate solution in $\mathbf{x}_0 + \mathbf{A}\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$, while MINRES extracts an approximate solution from $\mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$. The multiplication by $\mathbf{A}$ (of $\mathbf{r}_0$) has a demping effect on the components (of $\mathbf{r}_0$) in the direction of eigenvectors of $\mathbf{A}$ with small eigenvalues. The solution $\mathbf{y}$ of the system $\mathbf{A}\mathbf{y} = \mathbf{r}_0$ ($\mathbf{x} = \mathbf{x}_0 + \mathbf{y}$) often has large components in these directions. As a consequence, SYMMLQ tends to converge slower than MINRES (even though SYMMLQ aims for minimal errors): the extraction strategy of SYMMLQ is more to our liking, however, in general, its search subspace is of lower quality than for MINRES.

D (Nonlinear) CG as a minimisation method

There are several different ways to derive the CG algorithm. These different ways allow different generalisations that are effective for different types of problems.

We learnt that (1) CG can be obtained as an efficient variant of GCR (cf. Exercise 5.17). The approach in Exercise 7.3 (2) computes orthogonal residuals (cf., (7.4) with $\mathbf{M} = \mathbf{I}$) with coupled two-term recurrences. In Lecture 8, we will see that this leads to Bi-CG for efficient solution methods of general systems of equations (see Exercise 8.9), and from there to Bi-CGSTAB (see Exercise 8.11). In Exercise 7.5, we saw that (3) CG can be viewed as an efficient implementation of FOM, where symmetry allows to replace the Arnoldi relations by the three term recurrence relations of the Lanczos process (see Exercise 7.4). The ideas in this approach lead to MINRES and SYMMLQ for symmetric, but indefinite systems of equations (see Exercise 7.8 and Exercise 7.9). A fourth approach views (4) CG as a method to compute a minimum of a convex real valued function. This statement already indicates the type of problems to which this approach is generalised. We discuss this minimisation approach now.

Consider a real-valued map $\mathbf{F}$ on (a domain in) $\mathbb{R}^n$ that takes its minimum in $\mathbf{x}$. We want to compute $\mathbf{x}$. To guarantee that the minimum exists, $\mathbf{F}$ should be bounded from below and have some (local) convexity structure. We assume $\mathbf{F}$ to be sufficiently smooth; continuously differentiable for steepest descent and twice continuously differentiable for CG.

A linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ corresponds to the minimisation problem

$$\mathbf{x} = \mathrm{argmin}_{\widetilde{\mathbf{x}}} \tfrac{1}{2}\|\mathbf{b} - \mathbf{A}\widetilde{\mathbf{x}}\|^2_{A^{-1}} : \qquad \mathbf{F}(\mathbf{x}) \equiv \tfrac{1}{2}\|\mathbf{b} - \mathbf{A}\widetilde{\mathbf{x}}\|^2_{A^{-1}}. \tag{7.12}$$

We will see that CG can be viewed as a method that solves a *mini minimisation* in each step: it solves a second order approximate minimisation problem that is projected onto a two dimensional space to obtain a search direction, i.e., the direction into which the approximate

solution is updated. CG improves on the steepest descent method, where in each step the approximate solution $\widetilde{\mathbf{x}}$ is improved by 'descending' from there in the 'steepest' direction, i.e, in the direction that locally (close to $\widetilde{\mathbf{x}}$) gives the strongest decrease of $\mathbf{F}$-values. CG includes the steepest direction in the two dimensional space as well as the search direction from the preceding step.

We will exploit the Taylor expansions

$$\mathbf{F}(\widetilde{\mathbf{x}}+\mathbf{e}) = \mathbf{F}(\widetilde{\mathbf{x}}) + (\mathbf{e}, \nabla\mathbf{F}(\widetilde{\mathbf{x}})) + \mathcal{O}(\|\mathbf{e}\|_2^2) \qquad (\|\mathbf{e}\|_2 \to 0)$$

and

$$\mathbf{F}(\widetilde{\mathbf{x}}+\mathbf{e}) = \mathbf{F}(\widetilde{\mathbf{x}}) + (\mathbf{e}, \nabla\mathbf{F}(\widetilde{\mathbf{x}})) + \tfrac{1}{2}(\mathbf{H}(\widetilde{\mathbf{x}})\mathbf{e}, \mathbf{e}) + \mathcal{O}(\|\mathbf{e}\|_2^3) \qquad (\|\mathbf{e}\|_2 \to 0). \tag{7.13}$$

The n-vector $\nabla\mathbf{F}(\widetilde{\mathbf{x}})$ is the **gradient** of $\mathbf{F}$ at $\widetilde{\mathbf{x}}$ and the symmetric $n \times n$ matrix $\mathbf{H}(\widetilde{\mathbf{x}})$ is the **Hessian** of $\mathbf{F}$ at $\widetilde{\mathbf{x}}$. The n-vector $-\nabla\mathbf{F}(\widetilde{\mathbf{x}})$ gives the direction of the **steepest descent**: if $\mathbf{u}$ is the normalized gradient, then for any other direction vector $\widetilde{\mathbf{u}}$, i.e., $\widetilde{\mathbf{u}}$ is normalized, we have

$$\mathbf{F}(\widetilde{\mathbf{x}} - \delta\mathbf{u}) < \mathbf{F}(\mathbf{x} - \delta\widetilde{\mathbf{u}}) \quad \text{for all } \delta > 0 \text{ and sufficiently small.} \tag{7.14}$$

In case of (7.12), $-\widetilde{\mathbf{r}}$ with $\widetilde{\mathbf{r}} \equiv \mathbf{b} - \mathbf{A}\widetilde{\mathbf{x}}$ is the gradient, i.e., $\widetilde{\mathbf{r}}$ is the steepest descent direction:

Exercise 7.10.

(a) Prove (7.14).

(b) Show that $-\nabla\mathbf{F}(\widetilde{\mathbf{x}}) = \widetilde{\mathbf{r}} \equiv \mathbf{b} - \mathbf{A}\widetilde{\mathbf{x}}$ and $\mathbf{H}(\widetilde{\mathbf{x}}) = \mathbf{A}$ if $\mathbf{F}$ is as in (7.12):

$$\|\mathbf{b} - \mathbf{A}(\widetilde{\mathbf{x}}+\mathbf{e})\|_{A^{-1}}^2 = \|\widetilde{\mathbf{r}} - \mathbf{A}\mathbf{e}\|_{A^{-1}}^2 = \|\widetilde{\mathbf{r}}\|_{A^{-1}}^2 - 2(\widetilde{\mathbf{r}}, \mathbf{e}) + (\mathbf{e}, \mathbf{A}\mathbf{e}) \quad (\mathbf{e} \in \mathbb{R}^n).$$

(c) Compute the gradient and Hessian (in terms of $\widetilde{\mathbf{r}}$ and $\mathbf{A}$) in case

$$\mathbf{F}(\widetilde{\mathbf{x}}) \equiv \tfrac{1}{2}\|\mathbf{b} - \mathbf{A}\widetilde{\mathbf{x}}\|_2^2 \qquad (\widetilde{\mathbf{x}} \in \mathbb{R}^n). \tag{7.15}$$

The **steepest descent method** updates in each step an approximate solution $\widetilde{\mathbf{x}}$ in the direction $\widetilde{\mathbf{r}}$ of steepest descent, i.e., minus the gradient. It computes the optimal steplength in that direction, that is, a scalar α such that

$$\alpha = \text{argmin}\{\mathbf{F}(\widetilde{\mathbf{x}} + \alpha\widetilde{\mathbf{r}}) \,\big|\, \widetilde{\alpha} \in \mathbb{R}\},$$

and updates the approximate solution with $\mathbf{u} = \alpha\,\widetilde{\mathbf{r}}:\ \widetilde{\mathbf{x}} \leftarrow \widetilde{\mathbf{x}} + \alpha\widetilde{\mathbf{r}}$.

Non-linear CG exploits the second order approximation of $\mathbf{F}$ at $\widetilde{\mathbf{x}}$ (cf., (7.13)). It finds an update direction $\mathbf{u}$ from a two-dimensional space $\mathcal{V}$ by computing the minimiser from $\mathcal{V}$ for this second order approximation:

$$\mathbf{u} = \text{argmin}\{-2(\widetilde{\mathbf{r}}, \mathbf{v}) + \mathbf{v}^*\mathbf{H}(\mathbf{x})\mathbf{v} \,\big|\, \mathbf{v} \in \mathcal{V}\}. \tag{7.16}$$

The space $\mathcal{V}$ is spanned by the steepest descent direction plus the update vector from the preceding step. If $\mathbf{V}$ spans $\mathcal{V}$, then $\mathbf{u}$ solves (7.16) if and only if $\mathbf{u} = \mathbf{V}\vec{\beta}$ with $\vec{\beta}$ the 2-vector that solves

$$\mathbf{V}^*\mathbf{H}(\mathbf{x})\mathbf{V}\vec{\beta} = \mathbf{V}^*\widetilde{\mathbf{r}}. \tag{7.17}$$

The resulting algorithm is displayed in Alg. 7.5. Note that in non-linear CG the determination of the step-length α is redundant in case $\mathbf{F}$ is quadratic (then $\alpha = 1$). A stop-criterion has not been included in these algorithms. If $\mathbf{F}$ is as in (7.15), then $\mathbf{F}(\widetilde{\mathbf{x}})$ is a half times the square of the residual norm. Stopping if $\mathbf{F}(\widetilde{\mathbf{x}})$ is gives an accurate solution. In case of (7.12) (as for CG), $\mathbf{F}$ is the A-norm of the error and can not (readily) be computed. But in this case $\|\widetilde{\mathbf{r}}\|_2$ can be computed and can be used to monitor progress of the computation. If $\mathbf{F}(\widetilde{\mathbf{x}})$ can easily be computed, the decrease of this quantity can be monitored and may give information on the quality of the approximate solution (no decrease at full accuracy).

STEEPEST DESCENT

Select $\mathbf{x}_0 \in \mathbb{R}^n$

$\mathbf{x} = \mathbf{x}_0$

repeat

  %% *steepest descent direction*

  $\mathbf{r} = -\nabla \mathbf{F}(\mathbf{x})$

  %% *compute step-length*

  $\alpha = \mathtt{argmin}_{\widetilde{\alpha}} \mathbf{F}(\mathbf{x} + \widetilde{\alpha}\,\mathbf{r})$

  $\mathbf{x} \leftarrow \mathbf{x} + \alpha \mathbf{r}$ %% *update*

end repeat

NON-LINEAR CG

Select $\mathbf{x}_0 \in \mathbb{R}^n$

$\mathbf{x} = \mathbf{x}_0$, $\mathbf{u} = \mathbf{0}$

repeat

  $\mathbf{r} = -\nabla \mathbf{F}(\mathbf{x})$

  $\mathbf{V} = [\mathbf{r}, \mathbf{u}]$ %% *search matrix*

  Solve $\mathbf{V}^* \mathbf{H}(\mathbf{x}) \mathbf{V} \vec{\beta} = \mathbf{V}^* \mathbf{r}$ for $\vec{\beta}$

  $\mathbf{u} \leftarrow \mathbf{V}\vec{\beta}$ %% *CG direction*

  $\alpha = \mathtt{argmin}_{\widetilde{\alpha}} \mathbf{F}(\mathbf{x} + \widetilde{\alpha}\,\mathbf{u})$

  $\mathbf{x} \leftarrow \mathbf{x} + \alpha \mathbf{u}$

end repeat

ALGORITHM 7.5. Steepest descent and non-linear CG for computing the minimizer $\mathbf{x}$ of $\mathbf{F}$, $\mathbf{x} = \mathrm{argmin}\{\mathbf{F}(\mathbf{y})\}$, where $\mathbf{F}$ is a real valued function on $\mathbb{R}^n$. The left panel is steepest descend, the right one displays non-linear CG.

For many 'weakly non-quadratic' functions $\mathbf{F}$, the Hessian can easily be computed. Note also that the Hessian $\mathbf{H}$ at $\widetilde{\mathbf{x}}$ is not required, only the action on the two column vectors of $\mathbf{V}$.

Exercise 7.11.

(a) Prove the equivalence of (7.16) and (7.17).

(b) Show that in case of (7.12), the non-linear CG algorithm in ALG. 7.5 is equivalent to the CG algorithm that we saw before (with CG coefficients α and β absorbed in $\vec{\beta}$). Show that in this is also the case if we take $\mathbf{V} = [\mathbf{r}, \mathbf{x}, \mathbf{x}_{\mathrm{old}}]$ as search matrix in the non-linear CG algorithm. Here, $\mathbf{x}$ is the current approximate solution in non-linear CG and $\mathbf{x}_{\mathrm{old}}$ is the preceeding one.

(c) Show that non-linear CG for the function in (7.15) is equivalent to CR.

(d) Show that steepest descent for the function in (7.15) is equivalent to LMR.

E Optimal short recurrence methods.

[*This section is based on a report by Casper Beentjes, feb., 2014.*] Hermitian matrices allow an orthonormal basis of a Krylov subspace to be computed using short recurrences (at least in exact arithmetic): Lanczos relies on a three term recurrence relation to compute the Arnoldi relation $\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\underline{H}_k$, in which case $\underline{T}_k \equiv \underline{H}_k$ is tridiagonal. The following exercise shows that for some type of normal matrices the Arnoldi relation can also be determined with short recurrence relations.

First we observe that, for $s \in \mathbb{N}$, the Arnoldi relation can be determined with $(s+2)$-term recurrences if and only if the Hessenberg matrix $\underline{H}_k = (h_{ij})$ is s-banded, that is $h_{ij} = 0$ is $|i-j| > s$. Therefore, in the next exercise, it suffices to show that $\underline{H}_k$ is s-banded.

Exercise 7.12. Let $\mathbf{A}$ be a real $n \times n$ matrix (not necessarily Hermitian). Suppose there is a polynomial P of degree s such that $\mathbf{A}^* = P(\mathbf{A})$ (see Th. 0.16 and subsequent exercise).

(a) Prove that $\mathbf{A}$ is normal (i.e., $\mathbf{A}\mathbf{A}^* = \mathbf{A}^*\mathbf{A}$).

(b) Determine an appropriate polynomial in case $\mathbf{A}$ is Hermitian, and also one in case $\mathbf{A} = \alpha\mathbf{I} + \beta\mathbf{M}$ for some $\alpha, \beta \in \mathbb{C}$ and an Hermitian matrix $\mathbf{M}$.

(c) Show that the assumption is correct if $\mathbf{A}$ is normal and all eigenvalues are real except for at most s complex ones.

(d) Let $\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{H}$ be the 'maximal' Arnoldi relation (i.e., $\mathbf{V} = \mathbf{V}_m$ spans an m-dimensional subspace that is invariant under multiplication by $\mathbf{A}$ and m is as small as possible with give first

column $\mathbf{v}_1$. Note that $m = n$ gives an $\mathbf{A}$-invariant subspace). Show that $\mathbf{H}$ is $(s+1)$-banded. Conclude that $\underline{H}_k$ is $(s+1)$-banded for all $k \leq m$.

Faber and Manteuffel proved in a SINUM 1984 paper that the $\mathbf{H}$ of the maximal Arnoldi relation is $(s+1)$-banded if and only if there is a polynomial P of degree at most s such that $\mathbf{A}^* = P(\mathbf{A})$. The practical consequence of this result is that we only can work with short recurrences to form the Arnoldi relation for matrices of the form $\mathbf{A} = \alpha\mathbf{I} + \beta\mathbf{H}$ with $\mathbf{H}$ Hermitian: the more general permissible matrices are hard to identify in practise. Note that all eigenvalues for such a shifted and scaled Hermitian matrix are on one straight line in the complex plain; they are *collinear*.

One can consider to generalise the idea of an Arnoldi relation to

$$\mathbf{A}\mathbf{V}_k R_k = \mathbf{V}_{k+1}\,\underline{H}_k, \tag{7.18}$$

where R_k is an $k \times k$ non-singular upper triangular matrix that, like $\underline{H}_k$, is extended by one row and one column if k is increases by 1. As before, the matrices $\mathbf{V}_k$ are orthonormal.

Exercise 7.13. Consider (7.18).

(a) Show that the columns of $\mathbf{V}_k$ form an orthonormal Krylov basis generated by $\mathbf{A}$ and $\mathbf{v}_1$.

(b) Derive a GMRES type approach to solve $\mathbf{A}\mathbf{x} = \mathbf{b}$ based on the generalised Arnoldi relation (assuming we know the relation exists and we know how to extend the relation for each k).

If both R_k and $\underline{H}_k$ are s-banded then the generalised Arnoldi recursion can be generated by short recurrences (involving not only some 'previous' $\mathbf{v}_j$ to compute $\mathbf{v}_{k+1}$ but also $\mathbf{A}\mathbf{v}_j$), and efficiently leads to an orthonormal Krylov basis.

The next exercise shows that, in case $\mathbf{A} = \mathbf{Q}$ is unitary, we can easily form a generalised Arnoldi relation with both R_k and $\underline{H}_k$ bi-diagonal.

Exercise 7.14. Let $\mathbf{Q}$ be $n \times n$ unitary.

(a) Suppose $\mathbf{v}_1, \ldots, \mathbf{v}_k$ is a sequence of orthonormal vectors. We define $\mathbf{v}_{k+1}$ by the following relation. Show there are scalars γ_k, α_k, and β_k such that

$$\beta_k \mathbf{v}_{k+1} = \mathbf{Q}\mathbf{v}_k - \gamma_k \mathbf{Q}\mathbf{v}_{k-1} - \alpha_k \mathbf{v}_k \perp \mathbf{v}_k, \mathbf{v}_{k-1}, \quad \text{and} \quad \|\mathbf{v}_{k+1}\|_2 = 1. \tag{7.19}$$

We assume that $\mathbf{v}_2, \ldots, \mathbf{v}_k$ have been constructed with a formula as (7.19), $\mathrm{span}(\mathbf{V}_k) = \mathcal{K}_k(\mathbf{Q}, \mathbf{v}_1)$ (induction), and that $\mathrm{span}(\mathbf{V}_j)$ is not $\mathbf{Q}$-invariant for any $j \leq k$.

(b) Show that the non-$\mathbf{Q}$-invariance implies that

$$\mathbf{Q}^*\mathbf{v}_j \in \mathrm{span}(\mathbf{v}_1, \ldots, \mathbf{v}_{k-2}, \mathbf{Q}^*\mathbf{v}_{k-1}) \quad \text{for all } j = 1, 2, \ldots, k-1.$$

(c) Prove that $\mathbf{v}_k - \gamma_k \mathbf{v}_{k-1} \perp \mathbf{Q}^*\mathbf{v}_j$ for all $j = 1, 2, \ldots, k-1$.

(d) Conclude that $\beta_k \neq 0$, $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{k+1}$ is an orthonormal basis of $\mathcal{K}_{k+1}(\mathbf{Q}, \mathbf{v}_1)$, and

$$\mathbf{Q}\mathbf{V}_k(I_k + \Gamma_k) = \mathbf{V}_{k+1}\,\underline{B}_k$$

with Γ_k the $k \times k$ matrix with all entries equal to 0, except for the entries $\gamma_2, \ldots, \gamma_k$ on the first upper co-diagonal, and $\underline{B}_k$ the bi-diagonal matrix with $\alpha_1, \ldots, \alpha_k$ on the main diagonal and $\beta_1, \ldots, \beta_k$ and the first lower co-diagonal.

(e) Give an efficient algorithm to compute this generalised Arnoldi relation (how many multiplications by $\mathbf{Q}$ are required per step?).

(f) Let $\mathbf{A} = \sigma\,\mathbf{I} + \mu\,\mathbf{Q}$ for some scalars $\sigma, \mu \in \mathbb{C}$. We are interested in a short recurrence method for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$. We assume that $\mu = 1$. Why is this not a restriction? Show that

$$\mathbf{A}\mathbf{V}(I_k + \Gamma_k) = \mathbf{V}_{k+1}\,(\mu\,\underline{B}_k + \sigma\,\underline{I}_k + \sigma\,\underline{\Gamma}_k).$$

Derive a (MINRES type of) algorithm for such an iterative method such that the kth residual equals the kth GMRES residual (assuming $\mathbf{x}_0 = \mathbf{0}$ for both the new algorithm and for GMRES).

Barth and Manteuffel discussed the question (in a SIMAX 2000 paper) for what type of matrices there is a generalised Arnoldi relation with s-banded matrices R_k and $\underline{H}_k$. The central property of a matrix $\mathbf{A}$ in their discussion is $\mathbf{A}^*Q(\mathbf{A}) = P(\mathbf{A})$ for some polynomials P and Q that are relative prime. According to Cayley–Hamilton's theorem, $\mathbf{A}^{-1}$ can be represented as a polynomial in $\mathbf{A}$. Therefore, if $\mathbf{A}^*$ can be expressed as a rational function in $\mathbf{A}$, then it can be expressed as a polynomial in $\mathbf{A}$. However, using a rational function P/Q rather than a polynomial may allow to use polynomials of low degree.[2] For instance, $\mathbf{A}^*\mathbf{A} = \mathbf{I}$ if $\mathbf{A}$ is unitary, whence $Q(\lambda) = \lambda$ and $P(\lambda) = 1$, while $\mathbf{A}^* = P(\mathbf{A})$ may require a polynomial P if degree $n-2$ ($\mathbf{S}\mathbf{e}_j \equiv \mathbf{e}_{j-1}$ for $j = 2, \ldots, n$ and $\mathbf{S}\mathbf{e}_1 \equiv \mathbf{e}_n$: $\mathbf{S}^{n-1} = \mathbf{I}$ and $\mathbf{S}^{n-2} = \mathbf{S}^*$). The property $\mathbf{A}^*Q(\mathbf{A}) = P(\mathbf{A})$ allows to extend the result in (d) of Exercise 7.12.

For practical purposes, the only matrices for which a generalised Arnoldi relation can be formed using short recurrences are the shifted and scaled unitary matrices. Note that all eigenvalues of such a matrix are on one circle in the complex plane; they are *colcyclic*.

The inspiration to generalise the Arnoldi relation in order to find a larger class of matrices that allow short recurrences for computing an orthonormal Krylov basis comes from a 1982 paper by Gragg. Gragg derived the Isometric Arnoldi Process (IAP), an implementation of a coupled two term recurrence relation to compute the Arnoldi basis (orthonormal Krylov basis) in case the matrix $\mathbf{A}$ is unitary. His derivation is based on an analysis of the Hessenberg matrix $\mathbf{H}$ of the standard Arnoldi process using the fact that $\mathbf{H}$ is unitary as well. The IAP relates to the three term recurrence in (7.19) as CG relates to Lanczos. A shifted version of IAP combined with a MINRES type of approach led (Jagels and Reichel, 1994) to the Shifted Unitary Minimal Residual (SUMR) algorithm for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ with $\mathbf{A}$ shifted and scaled unitary. SUMR is equivalent to the algorithm indicated in Exercise 7.14(f).

In the above discussion, we assumed the standard inner product, $(\mathbf{x}, \mathbf{y}) = \mathbf{y}^*\mathbf{x}$. However, the discussion can easily be extended to any other inner product. For instance, if $\mathbf{M}$ is $n \times n$ positive definite, then

$$(\mathbf{x}, \mathbf{y})_M \equiv \mathbf{y}^*\mathbf{M}\mathbf{x} \qquad (\mathbf{x}, \mathbf{y} \in \mathbb{C}^n)$$

also defines an inner product, the M-inner product. With respect to the M-inner product, the adjoint $\mathbf{A}^*$ of a matrix $\mathbf{A}$ is such that $(\mathbf{A}\mathbf{x}, \mathbf{y})_M = (\mathbf{x}, \mathbf{A}^*\mathbf{y})_M$ $(\mathbf{x}, \mathbf{y} \in \mathbb{C}^n)$, whence $\mathbf{A}^* = \mathbf{M}^{-1}\mathbf{A}^{\mathrm{H}}\mathbf{M}$, where $\mathbf{A}^{\mathrm{H}}$ is the transpose complex conjugate of the matrix $\mathbf{A}$ (the adjoint of $\mathbf{A}$ with respect to the standard inner product). Then the central properties in the discussion on the computation of M-orthonormal Krylov basis with short recurrences will be 'self-adjointness', $\mathbf{A} = \mathbf{A}^*$, 'normality', $\mathbf{A}\mathbf{A}^* = \mathbf{A}^*\mathbf{A}$ and 'unitarity', $\mathbf{A}^*\mathbf{A} = \mathbf{I}$. Orthogonality as in (7.19) will be orthogonality with respect to the M-inner product. For computational efficiency, it will be convenient to store not only $\mathbf{v}_1, \ldots, \mathbf{v}_k$ but also $\mathbf{M}\mathbf{v}_1, \ldots, \mathbf{M}\mathbf{v}_k$ (that is, per step only the last few vectors of these sequences).

[2] That rational functions can be identified with polynomials in our discussion here comes from the fact that, the value of these functions is of importance only on the spectrum of $\mathbf{A}$, which contains only finitely many complex numbers.