March 26, 2018

Lecture 5 – Basic Iterative Methods II

The following two exercises show that eigenvalue problems and linear systems of equations are closely related. From a certain point of view, they are the same once the eigenvalue has been detected. Detection of the eigenvalue makes the eigenvalue problem (weakly) non-linear. Usually, when using iterative methods for solving these problems, convergence of the approximate eigenvalues is much faster than convergence of the eigenvectors: eigenvalues are detected before the associated eigenvectors.

Exercise 5.1. Let $\mathbf{A}$ be an $n \times n$ with rank $n-1$. Note that the vector that spans the null space of $\mathbf{A}$ is an eigenvector with eigenvalue 0. Here, we will see that the problem of finding this vector can be reformulated as problem of solving a linear system.

(a) Let $\mathbf{A}'$ be the matrix that arises by replacing, say, row ℓ of $\mathbf{A}$ by some row vector of dimension n such that $\mathbf{A}'$ is singular. Prove that the solution $\mathbf{x}$ of $\mathbf{A}'\mathbf{x} = \mathbf{e}_\ell$ spans the null space of $\mathbf{A}$.

(b) Select an ℓ. Let, in MATLAB notation, $I \equiv [1:\ell-1, \ell+1:n]$ and $\mathbf{A}' \equiv \mathbf{A}(I, I)$. Construct a vector that spans the null space of $\mathbf{A}$ from the solution of the equation $\mathbf{A}'\mathbf{x} = \mathbf{A}(I, \ell)$.

Exercise 5.2. Let $\mathbf{A}$ be a non-singular $n \times n$ matrix and $\mathbf{b}$ an n-vector. Consider the matrix

$$\widetilde{\mathbf{A}} \equiv \begin{bmatrix} 0 & \mathbf{0}^* \\ -\mathbf{b} & \mathbf{A} \end{bmatrix}$$

(a) Show that the solution $\mathbf{x}$ of the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ can be obtained from an eigenvector (with eigenvalue 0) of $\widetilde{\mathbf{A}}$.

(b) Suppose there is an $\alpha_0 \in \mathbb{C}$ and an $\rho \in [0, 1)$ such that $|1 - \alpha_0\lambda| \leq \rho$ for all eigenvalues λ of $\mathbf{A}$. Apply the power method to the shifted matrix $\mathbf{I} - \alpha_0\widetilde{\mathbf{A}}$. Scale the resulting vectors $\mathbf{u}_k$ such that the first coordinates are 1 (i.e., $\mathbf{e}_1^*\mathbf{u}_k = 1$ rather than $\|\mathbf{u}_k\|_2 = 1$). Relate the generated vectors to the approximate solutions of the system $\mathbf{A}\mathbf{x} = \mathbf{b}$ as produced by Richardson with parameter α_0.

A Two term iterative methods for eigenvalue problems

With convergence to an eigenvector, we mean directional convergence (as explained in the introduction of Lecture 4A.)

It can not be controlled to what eigenpair Rayleigh quotient iteration (RQI) converges. RQI does not even have to converge (as we saw in Exercise 4.12). But if it converges, then it converges fast. From the following theorem we learn that convergence is cubic if the matrix is Hermitian. In this theorem, we select an initial shift and an initial approximate eigenvector. If the shift is close to some eigenvalue or the approximate eigenvector is in direction close to some eigenvector, then the errors (with appropriate scaling) decrease (at least) cubically.

Note that the initial shift need not be a Rayleigh quotient in the theorem. This formulation makes the theorem also applicable to other types of iteration (as displayed in ALG. 5.1).

For general matrices, RQI converges quadratically if it converges.

Theorem 5.1 (Convergence RQI) *Let $\mathbf{A}$ be an $n \times n$ Hermitian matrix.*
Let $(\lambda, \mathbf{x})$ be an eigenpair of $\mathbf{A}$ with distance γ from λ to the other eigenvalues of $\mathbf{A}$.
Let $\mathbf{u}_0$ be a normalised vector and ρ_0 be a scalar. Let $\mathbf{u}_1$ be such that

$$(\mathbf{A} - \rho_0\mathbf{I})\widetilde{\mathbf{u}}_1 = \mathbf{u}_0, \quad \mathbf{u}_1 = \frac{\widetilde{\mathbf{u}}_1}{\|\widetilde{\mathbf{u}}_1\|_2}, \quad \textit{and} \quad \rho_1 \equiv \frac{\mathbf{u}_1^*\mathbf{A}\mathbf{u}_1}{\mathbf{u}_1^*\mathbf{u}_1}. \tag{5.1}$$

Shift-and-Invert	Rayleigh Quotient I.	Dominant Pole Alg.
Select	Select	Select
$\mathbf{u}_0 \in \mathbb{C}^n,\ \vartheta_0 \in \mathbb{C}$	$\mathbf{u}_0 \in \mathbb{C}^n,\ \vartheta_0 \in \mathbb{C}$	$\mathbf{u}_0 \in \mathbb{C}^n,\ \vartheta_0 \in \mathbb{C}$
$\mathbf{u} = \mathbf{u}_0,\ \vartheta = \vartheta_0$	$\mathbf{u} = \mathbf{u}_0,\ \vartheta = \vartheta_0$	$\mathbf{u} = \mathbf{u}_0,\ \vartheta = \vartheta_0$
$\mathbf{r} = \mathbf{A}\mathbf{u} - \vartheta\mathbf{u}$	$\mathbf{r} = \mathbf{A}\mathbf{u} - \vartheta\mathbf{u}$	$\mathbf{r} = \mathbf{A}\mathbf{u} - \vartheta\mathbf{u}$
while $\|\mathbf{r}\|_2 > tol$ do	while $\|\mathbf{r}\|_2 > tol$ do	while $\|\mathbf{r}\|_2 > tol$ do
Solve	Solve	Solve
$(\mathbf{A} - \vartheta_0\mathbf{I})\widetilde{\mathbf{u}} = \mathbf{u}$	$(\mathbf{A} - \vartheta\mathbf{I})\widetilde{\mathbf{u}} = \mathbf{u}$	$(\mathbf{A} - \vartheta\mathbf{I})\widetilde{\mathbf{u}} = \mathbf{u}_0$
for $\widetilde{\mathbf{u}}$	for $\widetilde{\mathbf{u}}$	for $\widetilde{\mathbf{u}}$
$\mathbf{u} = \widetilde{\mathbf{u}}/\|\widetilde{\mathbf{u}}\|_2$	$\mathbf{u} = \widetilde{\mathbf{u}}/\|\widetilde{\mathbf{u}}\|_2$	$\mathbf{u} = \widetilde{\mathbf{u}}/\|\widetilde{\mathbf{u}}\|_2$
$\vartheta = \mathbf{u}^*\mathbf{A}\mathbf{u}$	$\vartheta = \mathbf{u}^*\mathbf{A}\mathbf{u}$	$\vartheta = \mathbf{u}^*\mathbf{A}\mathbf{u}$
$\mathbf{r} = \mathbf{A}\mathbf{u} - \vartheta\mathbf{u}$	$\mathbf{r} = \mathbf{A}\mathbf{u} - \vartheta\mathbf{u}$	$\mathbf{r} = \mathbf{A}\mathbf{u} - \vartheta\mathbf{u}$
end while	end while	end while
$\lambda = \vartheta,\ \mathbf{x} = \mathbf{u}$	$\lambda = \vartheta,\ \mathbf{x} = \mathbf{u}$	$\lambda = \vartheta,\ \mathbf{x} = \mathbf{u}$

Algorithm 5.1. Variants of shift and invert power method for computing an eigenpair $(\lambda, \mathbf{x})$ of a general square matrix $\mathbf{A}$ with residual accuracy *tol*. For an initial approximate eigenpair $(\vartheta_0, \mathbf{u}_0)$, these algorithms compute updated approximate eigenpairs $(\vartheta, \mathbf{u})$. The algorithm at the left, Wielandt iteration or shift-and-invert (S&I), keeps the shift ϑ_0 fixed in each step. The algorithm in the middle, Rayleigh quotient iteration (RQI), updates the approximate eigenvector as well the eigenvalue in each step. The algorithm at the right, dominant pole algorithm (DPA), keeps the right hand side vector fixed in each step.

Put

$$\alpha_i \equiv \frac{|\rho_i - \lambda|}{\gamma - |\rho_i - \lambda|} \quad \textit{and} \quad t_i \equiv \tan\angle(\mathbf{u}_i, \mathbf{x}) \qquad (i = 0, 1).$$

If $|\rho_0 - \lambda| < \gamma$, *then we have that*

$$\alpha_0 t_0 < 1 \quad \Rightarrow \quad \alpha_1 \leq (\alpha_0 t_0)^2, \quad t_1 \leq \alpha_0 t_0, \quad \alpha_1 t_1 \leq (\alpha_0 t_0)^3.$$

Iterating (5.1) for $i = 2, \ldots$ with ρ_0 and ρ_1 replaced by ρ_{i-1} and ρ_i, respectively, and $\mathbf{u}_0$ and $\mathbf{u}_1$ by $\mathbf{u}_{i-1}$ and $\mathbf{u}_i$ defines **Rayleigh Quotient Iteration** (**RQI**; the middle algorithm in Alg. 5.1): RQI updates both ρ_i as well as $\mathbf{u}_i$. Keeping ρ_0 fixed and replacing $\mathbf{u}_0$ and $\mathbf{u}_1$ by $\mathbf{u}_{i-1}$ and $\mathbf{u}_i$, respectively, is **Wielandt iteration**, also called **Shift-and-Invert iteration** (**S&I**; the algorithm at left in Alg. 5.1). S&I only updates $\mathbf{u}_i$. S&I converges linearly. It favours the eigenpair with eigenvalue closest to ρ_0. Keeping $\mathbf{u}_0$ fixed and updating ρ_i is a third variant, **Dominant pole algorithm** (**DPA**; the algorithm at the right in Alg. 5.1). It converges quadratically, favouring the eigenpair with eigenvector closest to $\mathbf{u}_0$.

Exercise 5.3. Consider the setting of Theorem 5.1.

(a) Show that the theorem implies that RQI converges cubically towards $(\lambda, \mathbf{x})$ if ρ_0 is close to λ or if $\mathbf{u}_0$ is directionally close to $\mathbf{x}$ (note that the theorem has a statement on the product of an 'error' in eigenvalue times the error in the eigenvector. Discuss the implications for the eigenvalue and eigenvector separately).

(b) Show that the theorem implies that S&I converges linearly towards $(\lambda, \mathbf{x})$ if ρ_0 is close to λ and $\mathbf{u}_0$ is directionally close to $\mathbf{x}$ ($t_{i+1} \leq \alpha_0 t_i$ all i).

(c) Show that the theorem implies that DPA converges quadratically towards $(\lambda, \mathbf{x})$ if ρ_0 is close to λ and $\mathbf{u}_0$ is directionally close to $\mathbf{x}$ ($\alpha_i t_0^2 \leq (\alpha_i t_0^2)^2$ all i).

(d) Note that the algorithms in Alg. 5.1 require a 'solve' of a shifted system in each step, but not a matrix vector multiplication: explain how $\mathbf{A}\mathbf{u}$ (or $\mathbf{A}\widetilde{\mathbf{u}}$) (to update ϑ and the residual $\mathbf{r}$) can be computed with an AXPY (vector update) only.

Exercise 5.4. ***Proof of Theorem*** 5.1**.** Let $\mathbf{A}$ be $n \times n$ Hermitian with eigenvalues λ_j and associated normalised eigenvectors $\mathbf{x}_j$.
For simplicity, we assume the eigenvalues to be simple.

Here, we will not prove the precise result of Theorem 5.1. But, with some less technical details, we will prove that if RQI converges then it converges cubically.

We are interested in an eigenvalue $\lambda = \lambda_{j_0}$ and associated eigenvector $\mathbf{x} = \mathbf{x}_{j_0}$. Put $\gamma \equiv \inf_{j \neq j_0} |\lambda_j - \lambda_{j_0}|$: γ is the **spectral gap**, that is, the gap between the 'wanted' eigenvalue λ_{j_0} and the other eigenvalues.
We analyse one step of RQI. With $\mathbf{u}_0$, let $\mathbf{u}_1$ be such that

$$(\mathbf{A} - \rho_0 \mathbf{I})\mathbf{u}_1 = \mathbf{u}_0, \quad \text{where} \quad \rho_0 \equiv \frac{\mathbf{u}_0^* \mathbf{A} \mathbf{u}_0}{\mathbf{u}_0^* \mathbf{u}_0}.$$

(a) Show that, except for scaling factors, the $\mathbf{u}_i$ can be written as $\mathbf{u}_i = \mathbf{x} + \mathbf{y}_i$ for some vectors $\mathbf{y}_i \perp \mathbf{x}$ (assuming $\mathbf{u}_i$ has a non trivial component in the direction of $\mathbf{x}$). Conclude that, for some scaling factor τ we have that

$$\tau(\mathbf{A} - \rho_0 \mathbf{I})(\mathbf{x} + \mathbf{y}_1) = \mathbf{x} + \mathbf{y}_0.$$

Note that $\|\mathbf{y}_0\|_2 = t \equiv \tan(\angle(\mathbf{x}, \mathbf{u}_0))$, $\frac{\|\mathbf{y}_0\|_2^2}{1+\|\mathbf{y}_0\|_2^2} = s^2 \equiv \sin^2(\angle(\mathbf{x}, \mathbf{u}_0))$, and the scaling factor of $\mathbf{u}_0$ does not affect the angle $\angle(\mathbf{x}, \mathbf{u}_0)$.
(b) Prove that

$$|\rho_0 - \lambda| = \frac{|\mathbf{u}_0^*(\mathbf{A} - \lambda \mathbf{I})\mathbf{u}_0|}{\mathbf{u}_0^* \mathbf{u}_0} \leq \frac{|\mathbf{y}_0^*(\mathbf{A} - \lambda \mathbf{I})\mathbf{y}_0|}{1 + \|\mathbf{y}_0\|_2^2} \leq (\lambda_{\max} - \lambda_{\min}) \frac{\|\mathbf{y}_0\|_2^2}{1 + \|\mathbf{y}_0\|_2^2}.$$

Hence,

$$\frac{|\rho_0 - \lambda|}{\lambda_{\max} - \lambda_{\min}} \leq \sin^2(\angle(\mathbf{x}, \mathbf{u}_0)) :$$

approximate eigenvalues tend to be much more accurate than approximate eigenvectors.
(c) Show that $\tau(\mathbf{A} - \rho_0 \mathbf{I})\mathbf{x} = \mathbf{x}$ and $\tau(\mathbf{A} - \rho_0 \mathbf{I})\mathbf{y}_1 = \mathbf{y}_0$. Conclude that $\tau = 1/(\lambda - \rho_0)$.
Note that $\mathbf{y}_0$ and $\mathbf{y}_1$ are in the span of all $\mathbf{x}_j$ with $j \neq j_0$. Show that

$$\|\mathbf{y}_0\|_2 = |\tau| \, \|(\mathbf{A} - \rho_0 \mathbf{I})\mathbf{y}_1\|_2 \geq |\tau| \, \widetilde{\gamma} \, \|\mathbf{y}_1\|_2,$$

where $\widetilde{\gamma} \equiv \gamma - |\rho_0 - \lambda|$. Hence, if $|\rho_0 - \lambda| < \gamma$ then

$$\|\mathbf{y}_1\|_2 \leq |\lambda - \rho_0| \, \frac{\|\mathbf{y}_0\|_2}{\widetilde{\gamma}} \leq \frac{\|\mathbf{y}_0\|_2^3}{1 + \|\mathbf{y}_0\|_2^2} \, \frac{\lambda_{\max} - \lambda_{\min}}{\widetilde{\gamma}}.$$

(d) Show that RQI convergences at least cubic if there is convergence: except for some constant, the angle between the approximate eigenvector and the eigenvector reduces by at least a power of three in each step. (Hint: put $C \equiv 2(\lambda_{\max} - \lambda_{\min})/\gamma$. Then $\|\mathbf{y}_1\|_2 \leq C \|\mathbf{y}_0\|_2^2$ if $|\rho_0 - \lambda| \leq \frac{1}{2}\gamma \, C \, \|\mathbf{y}_0\|_2^2$).

What can you tell about the convergence of the associated approximate eigenvalues?

B Two term iterative methods for solving linear systems

Let $\mathbf{A}$ be an $n \times n$ matrix and $\mathbf{b}$ and n-vector.

If p is the polynomial $p(\zeta) = \alpha_0 + \alpha_1 \zeta + \ldots + \alpha_k \zeta^k \quad (\zeta \in \mathbb{C})$, then $p(\mathbf{A})$ is the matrix $p(\mathbf{A}) \equiv \alpha_0 \mathbf{I} + \alpha_1 \mathbf{A} + \ldots + \alpha_k \mathbf{A}^k$.

Exercise 5.5. **Richardson.** Consider Richardson's method in Alg. 5.2.
(a) Prove that the kth residual $\mathbf{r}_k$ in Richardson equals $\mathbf{r}_k = p_k(\mathbf{A})\mathbf{r}_0$ for $p_k(\zeta) \equiv (1 - \alpha\zeta)^k$.

Assume the eigenvalues of $\mathbf{A}$ are contained in $\{\zeta \in \mathbb{C} \,\big|\, |\alpha_0 - \zeta| \leq \rho\}$ with $0 \leq \rho < |\alpha_0|$.
(b) Prove that $\|\mathbf{r}_k\|_2 \to 0$ for $k \to \infty$. Show that for large k, $\|\mathbf{r}_{k+1}\|_2 \lesssim \frac{\rho}{|\alpha_0|} \|\mathbf{r}_k\|_2$.

Richardson iteration
Select $\mathbf{x}_0 \in \mathbb{C}^n$, $\alpha_0 \in \mathbb{C}$
$\mathbf{x} = \mathbf{x}_0$, $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
while $\|\mathbf{r}\|_2 > tol$ do
$\mathbf{u} = \mathbf{r}$
$\mathbf{c} = \mathbf{A}\mathbf{u}$
$\alpha = \alpha_0$
$\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c}$
$\mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u}$
end while

Local Minimal Residuals
Select $\mathbf{x}_0 \in \mathbb{C}^n$
$\mathbf{x} = \mathbf{x}_0$, $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
while $\|\mathbf{r}\|_2 > tol$ do
$\mathbf{u} = \mathbf{r}$
$\mathbf{c} = \mathbf{A}\mathbf{u}$
$\alpha = \frac{\mathbf{c}^*\mathbf{r}}{\mathbf{c}^*\mathbf{c}}$
$\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c}$
$\mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u}$
end while

Algorithm 5.2. Two term iterations for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$ with residual accuracy *tol*. Upon termination, $\mathbf{x}$ is the approximate solution with residual $\mathbf{r}$, $\|\mathbf{r}\|_2 < tol$.
Note that both algorithms are identical except for the choice of α. Richardson iteration (at the left) is based on the assumption that all eigenvalues of the matrix $\mathbf{A}$ are contained in the disc $\{\zeta \in \mathbb{C} \,\big|\, |\alpha_0 - \zeta| \leq \rho\}$ with $0 \leq \rho < |\alpha_0|$ and α_0 is available. Local minimal residual (LMR, at the right), selects α to minimise $\|\mathbf{r} - \alpha\,\mathbf{c}\|_2$, or, equivalently, $\mathbf{r} - \alpha\,\mathbf{c} \perp \mathbf{c}$.

(c) Assume that $\mathbf{A}$ is Hermitian and all eigenvalues are in $[\alpha_0 - \rho, \alpha_0 + \rho] = [\lambda_-, \lambda_+] \subset (0, \infty)$. Prove that

$$\|\mathbf{r}_k\|_2 \leq \left(\frac{\lambda_+ - \lambda_-}{\lambda_- + \lambda_-}\right)^k \|\mathbf{r}_0\|_2 \leq \exp\left(-2k\frac{\lambda_-}{\lambda_+}\right) \|\mathbf{r}_0\|_2.$$

Exercise 5.6. Local minimal residual. Consider the LMR method in Alg. 5.2.

(a) Prove that the kth residual in LMR equals $\mathbf{r}_k = p_k(\mathbf{A})\mathbf{r}_0$ for $p_k(\zeta) \equiv (1-\alpha_1\zeta)\cdot\ldots\cdot(1-\alpha_k\zeta)$ where α_j is the α as computed in step j.

(b) Assume $\frac{1}{2}(\mathbf{A} + \mathbf{A}^*)$ is definite with absolute smallest eigenvalue σ. Prove that

$$\|\mathbf{r}_{k+1}\|_2^2 = \|\mathbf{r}_k\|_2^2 \left(1 - \frac{|\mathbf{r}_k^*\mathbf{A}\mathbf{r}_k|^2}{\|\mathbf{A}\mathbf{r}_k\|_2^2\,\|\mathbf{r}_k\|_2^2}\right) \leq \|\mathbf{r}_k\|_2^2 \left(1 - \frac{\sigma^2}{\|\mathbf{A}\|_2^2}\right).$$

(c) Prove that LMR does not break down (i.e., no division by 0) in case $\frac{1}{2}(\mathbf{A} + \mathbf{A}^*)$ is definite.

(d) Prove that LMR converges in case $\frac{1}{2}(\mathbf{A} + \mathbf{A}^*)$ is definite.

(e) Show that LMR need not to converge if $\frac{1}{2}(\mathbf{A} + \mathbf{A}^*)$ is indefinite.

(f) Prove that the statement in Exercise 5.5(c) is also correct for LMR.

C Residual polynomials

Let $\mathbf{A}$ be an $n \times n$ matrix. Let $\mathbf{b}$ a non-trivial n-vector.

The **Krylov subspace** $\mathcal{K}_{k+1}(\mathbf{A}, \mathbf{b})$ of **order** $k+1$ generated by $\mathbf{A}$ and the n-vector $\mathbf{b}$ is the subspace of $\mathbb{C}^n$ spanned by $\mathbf{b}, \mathbf{A}\mathbf{b}, \ldots, \mathbf{A}^k\mathbf{b}$:

$$\mathcal{K}_{k+1}(\mathbf{A}, \mathbf{b}) \equiv \text{span}\,(\mathbf{b}, \mathbf{A}\mathbf{b}, \ldots, \mathbf{A}^k\mathbf{b}) = \{p(\mathbf{A})\mathbf{b} \,\big|\, p \in \mathcal{P}_k\},$$

where $\mathcal{P}_k$ is the space of all polynomials of degree at most k.

The methods that we discussed for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$, **Krylov subspace method**s, find approximate solutions $\mathbf{x}_k$ in $\mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$,

$$\mathbf{x}_k \in \mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0),$$

POLYNOMIAL ITERATION

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

Select $\zeta_1, \ldots, \zeta_\ell$

$k = 1, \ \mathbf{x} = \mathbf{x}_0, \ \mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$

while $\|\mathbf{r}\|_2 > tol$ do

  $\mathbf{u} = \mathbf{r}$

  $\mathbf{c} = \mathbf{A}\mathbf{u}$

  $\alpha = 1/\zeta_k$

  $\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c}$

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

  $k \leftarrow k + 1$, if $k > \ell$, $k = 1$, end if

end while

ALGORITHM 5.3. Polynomial iteration: two term iteration with a fixed polynomial for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$ with residual accuracy *tol*. The ζ_j are zeros of a polynomial p of degree ℓ. The zeros have to be selected such that $|p(\lambda)| < |p(0)|$ for all eigenvalues λ of $\mathbf{A}$.

that is, $\mathbf{x}_k = \mathbf{x}_0 + \mathbf{y}_k$ for some $\mathbf{y}_k \in \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$. Here, $\mathbf{x}_0$ is some initial guess in $\mathbb{C}^n$ (as $\mathbf{x}_0 = \mathbf{0}$) and $\mathbf{r}_0 \equiv \mathbf{b} - \mathbf{A}\mathbf{x}_0$. In particular, $\mathbf{y}_k = q(\mathbf{A})\mathbf{r}_0$ for some $q \in \mathcal{P}_{k-1}$, and

$$\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k = \mathbf{r}_0 - \mathbf{A}\mathbf{y}_k = p_k(\mathbf{A})\mathbf{r}_0 \in \mathcal{K}_{k+1}(\mathbf{A}, \mathbf{r}_0),$$

where $p_k(\zeta) \equiv 1 - \zeta\, q(\zeta) \ \ (\zeta \in \mathbb{C})$. Note that $p_k \in \mathcal{P}_k$ and $p_k(0) = 1$. Conversely, if $p_k \in \mathcal{P}_k$ and $p_k(0) = 1$, then there is some polynomial $q \in \mathcal{P}_{k-1}$ such that $p_k(\zeta) \equiv 1 - \zeta\, q(\zeta) \ \ (\zeta \in \mathbb{C})$ and $p_k(\mathbf{A})\mathbf{r}_0 = \mathbf{r}_0 - \mathbf{A}(q(\mathbf{A})\mathbf{r}_0)$: $p_k(\mathbf{A})\mathbf{r}_0$ is a residual. Therefore, we call polynomials that are 1 in 0, **residual polynomials**. We put

$$\mathcal{P}_k^0 \equiv \{p \,\big|\, p \text{ is a polynomial of degree } \leq k,\ p(0) = 1\}$$

Conclusion: the residuals that we compute are of the form $\mathbf{r}_k = p_k(\mathbf{A})\mathbf{r}_0$ with $p_k \in \mathcal{P}_k^0$.
To simplify notation, we will often take $\mathbf{x}_0 = \mathbf{0}$.

We would like to have small residuals for the system $\mathbf{A}\mathbf{x} = \mathbf{b}$.
To estimate $\|\mathbf{r}_k\|_2$ for a residual $\mathbf{r}_k = p_k(\mathbf{A})\mathbf{r}_0 \in \mathcal{K}_{k+1}(\mathbf{A}, \mathbf{r}_0)$, note that, in case $\mathbf{A}$ is diagonalizable,

$$\|\mathbf{r}_k\|_2 \leq \mathcal{C}_E\ \nu_k(\Lambda(\mathbf{A}))\, \|\mathbf{r}_0\|_2, \quad \text{where} \quad \nu_k(\mathcal{G}) \equiv \max\{|p_k(\zeta)| \,\big|\, \zeta \in \mathcal{G}\} \quad (\mathcal{G} \subset \mathbb{C}),$$

$\Lambda(\mathbf{A})$ the spectrum of $\mathbf{A}$, and $\mathcal{C}_E \equiv \|\mathbf{V}\|_2 \|\, \|\mathbf{V}^{-1}\|_2$ is the conditioning of the (best conditioned) basis of eigenvectors: $\mathbf{V}$ is $n \times n$ such that $\mathbf{A}\mathbf{V} = \mathbf{V}\Lambda$. In particular, if $\mathbf{A}$ is normal (then there is an orthonormal basis of eigenvectors), then $\|\mathbf{r}_k\|_2 \leq \nu_k(\Lambda(\mathbf{A}))\, \|\mathbf{r}_0\|_2$ and the size of the residual is (essentially) determined by the size of the polynomial p_k on the spectrum $\Lambda(\mathbf{A})$.

To have small residuals, we would like to have residual polynomials that are as small as possible on the spectrum. The polynomial, that is the zeros of the polynomial, can be the input for an algorithm, as is the case in ALG. 5.3. The zeros have to be strategically selected in an area that contains the spectrum of $\mathbf{A}$. It assumes some information about the spectrum.

Exercise 5.7. In every ℓth step of ALG. 5.3, i.e., if $k = m\ell$, the residual $\mathbf{r}_k$ is given by

$$\mathbf{r}_{m\ell} = p(\mathbf{A})^m \mathbf{r}_0, \quad \text{where} \quad p(\zeta) \equiv (1 - \tfrac{\zeta}{\zeta_1}) \cdot \ldots \cdot (1 - \tfrac{\zeta}{\zeta_\ell}) \quad (\zeta \in \mathbb{C}).$$

Methods as GCR and GMRES determine the polynomials itself. Unfortunately, the steps of GCR and GMRES are expensive (the costs per step are proportional to the step number).

For methods as **Chebyshev iteration** costs per step are limited by providing the residual polynomial as input. Since the spectrum of the matrix will not be known, a subset $\mathcal{G}$ of $\mathbb{C}$ has to be estimated that contains all eigenvalues $\Lambda(\mathbf{A}) \subset \mathcal{G}$, and polynomials are obtained as solution of

$$\operatorname{argmin} \max\{|p_k(\zeta)| \,\big|\, \zeta \in \mathcal{G}\}, \tag{5.2}$$

where the minimum is taken over all $p_k \in \mathcal{P}_k^0$.

If the spectrum is known to be contained in some interval $[\lambda_-, \lambda_+]$ of $(0, \infty)$, with boundaries λ_- and λ_+ available, then a shifted and scaled version of the Chebyshev polynomial T_k solve the minimisation problem (5.2) for $\mathcal{G} = [\lambda_-, \lambda_+]$:

Theorem 5.2 *For the interval $[\lambda_-, \lambda_+]$ with centre $\mu \equiv \frac{1}{2}(\lambda_- + \lambda_+)$ and radius $\rho \equiv \frac{1}{2}(\lambda_+ - \lambda_-)$, the polynomial*

$$x \rightsquigarrow \frac{1}{T_k(\frac{\mu}{\rho})}\, T_k\Big(\tfrac{\mu - x}{\rho}\Big) \qquad (x \in \mathbb{C})$$

solves problem (5.2) *for $\mathcal{G} \equiv [\lambda_-, \lambda_+] = [\mu - \rho, \mu + \rho]$ and*

$$\nu_k(\mathcal{G}) = \max\left\{ \frac{1}{|T_k(\frac{\mu}{\rho})|} \left| T_k\Big(\tfrac{\mu - x}{\rho}\Big)\right| \,\Big|\, x \in \mathcal{G} \right\} = \frac{1}{|T_k(\frac{\mu}{\rho})|} \le 2 \exp\left(-2k \sqrt{\frac{\lambda_-}{\lambda_+}} \right).$$

Exercise 5.8. **Chebyshev polynomials.** Let $T_\ell(x) \equiv \frac{1}{2}(\zeta^\ell + \zeta^{-\ell})$ if $x = \frac{1}{2}(\zeta + \zeta^{-1})$ $(x, \zeta \in \mathbb{C})$.

(a) Prove that $T_0(x) = 1,\ T_1(x) = x,$

$$T_{k+1}(x) = 2\, x\, T_k(x) - T_{k-1}(x) \quad \text{for all } x \in \mathbb{C} \quad \text{and} \quad k = 1, 2, \ldots. \tag{5.3}$$

Conclude that T_ℓ is a polynomial of exact degree ℓ. T_ℓ is the ℓth **Chebyshev polynomial**.

(b) Show that $x \in [-1, +1]$ is of the form $x = \cos(\phi)$ and that $T_\ell(x) = \cos(\ell\phi)$. Conclude that

$$|T_\ell(x)| \le 1 \quad \text{for all } \ x \in [-1, +1].$$

Show that T_ℓ takes the values $(-1)^k$ at the $\ell + 1$ values $x_k = \cos(\pi k/\ell)$ $(k = 0, 1, \ldots, \ell)$.

(c) For $\varepsilon > 0$ and $y \equiv \frac{1+\varepsilon}{1-\varepsilon}$, consider a polynomial q of degree $\le \ell$ that is equal to T_ℓ at y. Show that $|q(x)| \ge 1$ for some $x \in [-1, +1]$:

$$q \in \mathcal{P}_\ell \quad \& \quad q(y) = T_\ell(y) \qquad \Rightarrow \qquad \max_{x \in [-1,+1]} |q(x)| \ge 1.$$

(Hint: If $|q(x)| < 1$ for all $x \in [-1, 1]$ then the graph of q intersects the graph of T_ℓ on $[-1, +1]$ in at least ℓ points (Why? Draw a picture for $\ell = 3$). Conclude that $q - T_\ell$ has at least $\ell + 1$ zeros. Argue that this is not possible.) Conclude that $T_\ell / T_\ell(y)$ is the 'smallest' polynomial of degree ℓ on $[-1, +1]$ (smallest $\max|\cdot|$ with maximum over $x \in [-1, +1]$) with value 1 at y.

(d) If $\zeta = \frac{1+\delta}{1-\delta}$, then $x = \frac{1}{2}(\zeta + \zeta^{-1}) = \frac{1+\delta^2}{1-\delta^2}$. Prove this and show that

$$|T_\ell(y)| \ge \frac{1}{2}\left(\frac{1 + \sqrt{\varepsilon}}{1 - \sqrt{\varepsilon}} \right)^\ell \ge \frac{1}{2} \exp\left(2\ell\sqrt{\varepsilon}\right).$$

Hence,

$$\frac{|T_\ell(x)|}{|T_\ell(y)|} \le 2 \exp\left(-2\ell\sqrt{\varepsilon}\right) \qquad (x \in [-1, +1]).$$

(e) Prove Theorem 5.2.

(f) Consider the subset $\mathcal{E}_\delta \equiv \{\frac{1}{2}(\zeta + \zeta^{-1}) \,\big|\, \zeta \in \mathbb{C}, \frac{1-\delta}{1+\delta} \le |\zeta| \le \frac{1+\delta}{1-\delta}\}$ of the complex plane. Show that this set defines the interior of an ellipse that contains $[-1, +1]$. If $\zeta \in \mathcal{E}_\delta$ then $\bar{\zeta} \in \mathcal{E}_\delta$. Prove that

$$\frac{|T_\ell(x)|}{|T_\ell(y)|} \le 2 \exp\Big(-2\ell(\sqrt{\varepsilon} - \sqrt{\delta})\Big) \qquad (x \in \mathcal{E}_\delta).$$

```
Select μ > ρ, ℓ ∈ ℕ
for k = 1, ..., ℓ,
   ζ_k = μ − ρ cos(π (2k−1)/(2ℓ))
end for
```

ALGORITHM 5.4. The selection of the zeros of a fixed shifted ℓth degree Chebyshev polynomial for use in ALG. 5.3. This selection is based on the assumption that all eigenvalues of the matrix $\mathbf{A}$ are contained in the interval $[\mu-\rho, \mu+\rho] \subset (0,\infty)$ and that both μ and ρ are available.

This result shows that the scaled and shifted Chebyshev polynomials are also small on the spectrum of $\mathbf{A}$ if the eigenvalues are in the ellipsoid $\mu - \rho\mathcal{E}_\delta$ (that does not contain 0).

Exercise 5.9. **Polynomial iteration with a fixed 'Chebyshev' polynomial.** Consider the situation as in Theorem 5.2.

(a) Show that the Chebyshev polynomial T_ℓ has zeros at $\cos\left(\pi\frac{2k-1}{2\ell}\right)$ $(k = 1, 2, \ldots, \ell)$. Conclude that the zeros ζ_k of the shifted and scaled Chebyshev polynomial of Theorem 5.2 are $\mu - \rho\cos\left(\pi\frac{2k-1}{2\ell}\right)$ for $k = 1, \ldots, \ell$.

(b) Prove that

$$\frac{1}{T_\ell(\frac{\mu}{\rho})}\,T_\ell\Big(\frac{\mu-\zeta}{\rho}\Big) = (1-\tfrac{\zeta}{\zeta_1})(1-\tfrac{\zeta}{\zeta_2})\cdot\ldots\cdot(1-\tfrac{\zeta}{\zeta_\ell}) \quad \text{with} \quad \zeta_k \equiv \mu - \rho\cos\left(\pi\tfrac{2k-1}{2\ell}\right).$$

(c) In every ℓth steps of polynomial iteration (Alg. 5.3) using the zeros of the shifted and scaled Chebyshev polynomial (of (b); see Alg. 5.4), we have for the residual $\mathbf{r}_k$

$$\|\mathbf{r}_k\|_2 \le \mathcal{C}_E\, 2^m \exp\left(-2k\sqrt{\frac{\lambda_-}{\lambda_+}}\right)\|\mathbf{r}_0\|_2 \qquad (k = m\ell, m = 0, 1, 2, \ldots)$$

(d) What is the effect on the convergence history, that is, on the norm of the residuals in the sequence $(\mathbf{r}_k)$ of changing the order of selecting the α_j, i.e., $\alpha_k = 1/(\mu + \rho\cos\left(\pi\frac{2k-1}{2\ell}\right))$ rather than $\alpha_k = 1/(\mu - \rho\cos\left(\pi\frac{2k-1}{2\ell}\right))$? Distinguish the cases $k = m\ell$ and $k \neq m\ell$.

Exercise 5.10. **Chebyshev iteration (three term recurrences).** We are interested in methods for the numerical solution of $\mathbf{x}$ from $\mathbf{Ax} = \mathbf{b}$. We derive an algorithm to compute approximate solutions $\mathbf{x}_k$ with residuals $\mathbf{r}_k$ with (cf., Theorem 5.2)

$$\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k = \frac{1}{\gamma_k}\mathbf{s}_k, \quad \text{where} \quad \mathbf{s}_k \equiv T_k\Big(\tfrac{1}{\rho}(\mu\mathbf{I} - \mathbf{A})\Big)\,\mathbf{r}_0 \quad \text{and} \quad \gamma_k \equiv T_k(\tfrac{\mu}{\rho}).$$

To simplify notation, we take $\mathbf{x}_0 = \mathbf{0}$, whence $\mathbf{r}_0 = \mathbf{b}$.

(a) Show that $\gamma_0 = 1, \quad \gamma_1 = \frac{\mu}{\rho}, \quad \gamma_{k+1} = 2\,\frac{\mu}{\rho}\,\gamma_k - \gamma_{k-1} \quad (k = 1, 2, \ldots)$ and

$$\mathbf{s}_0 = \mathbf{r}_0, \quad \mathbf{s}_1 = \tfrac{1}{\rho}\,(\mu\mathbf{r}_0 - \mathbf{A}\mathbf{r}_0), \quad \mathbf{s}_{k+1} = \tfrac{2}{\rho}\,(\mu\mathbf{s}_k - \mathbf{A}\mathbf{s}_k) - \mathbf{s}_{k-1} \quad (k = 1, 2, \ldots).$$

(b) Put $\zeta_k \equiv \frac{\gamma_k}{\gamma_{k-1}}$. Prove that $\zeta_1 = \frac{\mu}{\rho}, \quad \zeta_{k+1} = 2\frac{\mu}{\rho} - \frac{1}{\zeta_k}$, and

$$\mathbf{r}_1 = \mathbf{r}_0 - \tfrac{1}{\mu}\mathbf{A}\mathbf{r}_0, \quad \zeta_{k+1}\,\mathbf{r}_{k+1} = \tfrac{2}{\rho}(\mu\mathbf{r}_k - \mathbf{A}\mathbf{r}_k) - \tfrac{1}{\zeta_k}\,\mathbf{r}_{k-1} \quad (k = 1, 2, \ldots). \tag{5.4}$$

(c) Use an induction argument to show that the $(\mathbf{x}_k)$ satisfy

$$\mathbf{x}_1 = \tfrac{1}{\mu}\mathbf{r}_0, \quad \zeta_{k+1}\,\mathbf{x}_{k+1} = \tfrac{2}{\rho}(\mu\mathbf{x}_k + \mathbf{r}_k) - \tfrac{1}{\zeta_k}\,\mathbf{x}_{k-1} \quad (k = 1, 2, \ldots). \tag{5.5}$$

CHEBYSHEV IT. (3-TERM)	CHEBYSHEV I. (COUPLED 2-TERM)
Select $\mu > \rho > 0$, $\mathbf{x}_0 = \mathbf{0}$	Select $\mu > \rho > 0$, $\mathbf{x}_0 \in \mathbb{C}^n$
$\zeta = \frac{\mu}{\rho}$, $\mathbf{r}_0 = \mathbf{b}$	$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
$\mathbf{x} = \frac{1}{\mu}\mathbf{r}_0$, $\mathbf{r} = \mathbf{r}_0 - \mathbf{A}\mathbf{x}$	$\mathbf{u} = \mathbf{r}_0$, $\mathbf{c} = \mathbf{A}\mathbf{u}$, $\zeta = \frac{\mu}{\rho}$
while $\|\mathbf{r}\|_2 > tol$ do	$\mathbf{r} = \mathbf{r}_0 - \frac{1}{\mu}\mathbf{c}$, $\mathbf{x} = \mathbf{x}_0 - \frac{1}{\mu}\mathbf{u}$
$\mathbf{s} = \frac{2}{\rho}(\mu\mathbf{r} - \mathbf{A}\mathbf{r}) - \frac{1}{\zeta}\mathbf{r}_0$	while $\|\mathbf{r}\|_2 > tol$ do
$\mathbf{y} = \frac{2}{\rho}(\mu\mathbf{x} + \mathbf{r}) - \frac{1}{\zeta}\mathbf{x}_0$	$\beta = \frac{1}{\zeta^2}$
$\mathbf{r}_0 = \mathbf{r}$	$\mathbf{u} \leftarrow \mathbf{r} + \beta\mathbf{u}$
$\mathbf{x}_0 = \mathbf{x}$	$\mathbf{c} = \mathbf{A}\mathbf{u}$
$\zeta \leftarrow 2\frac{\mu}{\rho} - \frac{1}{\zeta}$	$\zeta \leftarrow 2\frac{\mu}{\rho} - \frac{1}{\zeta}$, $\alpha = \frac{2}{\zeta\mu}$
$\mathbf{r} = \frac{1}{\zeta}\mathbf{s}$	$\mathbf{r} \leftarrow \mathbf{r} - \alpha\mathbf{c}$
$\mathbf{x} = \frac{1}{\zeta}\mathbf{y}$	$\mathbf{x} \leftarrow \mathbf{x} + \alpha\mathbf{u}$
end while	end while

ALGORITHM 5.5. Chebyshev iteration for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$ with residual accuracy *tol*. The algorithm is based on the assumption that all eigenvalues of the matrix $\mathbf{A}$ are contained in the interval $[\mu - \rho, \mu + \rho] \subset (0, \infty)$ and that both μ and ρ are available. At the left we have the three term variant and at the right the two coupled two term variant.

(d) Note that old quantities ($\mathbf{x}_j$ and $\mathbf{r}_j$ for $j < k-1$) are not needed anymore to compute $\mathbf{x}_{k+1}$ and $\mathbf{r}_{k+1}$. Derive the left algorithm in ALG. 5.5. It is possible to save $2n$ flop (floating point operation) per step (i.e., two scalar vector multiplications). How?

(e) Show that every residual $\mathbf{r}_k$ equals

$$\mathbf{r}_k = \frac{1}{T_k(\frac{\mu}{\rho})} T_k\Big(\tfrac{1}{\rho}(\mu\mathbf{I} - \mathbf{A})\Big)\,\mathbf{b}, \quad \|\mathbf{r}_k\|_2 \le \mathcal{C}_E\, 2\exp\left(-2k\sqrt{\frac{\lambda_-}{\lambda_+}}\right)\|\mathbf{b}\|_2.$$

(f) Compare (convergence, computational costs per step) the method in ALG. 5.5 (left) (of this exercise) with the one in ALG. 5.4 (of Exercise 5.9).

Exercise 5.11. **Chebyshev iteration (coupled two term recurrences).** This exercises continues the preceding one.

The process in in the previous exercise is a three term iteration: the new residual $\mathbf{r}_{k+1}$ is defined by three vectors (two preceding residuals $\mathbf{r}_k$ and $\mathbf{r}_{k-1}$ plus $\mathbf{A}\mathbf{r}_k$). Here we will derive a two coupled two term recurrence process.

(a) Use $\zeta_{k+1} = 2\frac{\mu}{\rho} - \frac{1}{\zeta_k}$ to show that

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \tfrac{2}{\rho\zeta_{k+1}}\Big(\mathbf{A}\mathbf{r}_k + \tfrac{\rho}{2\zeta_k}(\mathbf{r}_{k-1} - \mathbf{r}_k)\Big)$$

(b) Assume $\mathbf{u}_k$ is such that $\mathbf{x}_k = \mathbf{x}_{k-1} + \frac{2}{\rho\zeta_k}\mathbf{u}_{k-1}$. Show that $\mathbf{r}_{k-1} - \mathbf{r}_k = \frac{2}{\rho\zeta_k}\mathbf{A}\mathbf{u}_{k-1}$.

(c) Conclude that

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \tfrac{2}{\rho\zeta_{k+1}}\mathbf{A}\mathbf{u}_k, \quad \mathbf{x}_{k+1} = \mathbf{x}_k + \tfrac{2}{\rho\zeta_{k+1}}\mathbf{u}_k, \quad \text{where} \quad \mathbf{u}_k = \mathbf{r}_k + \tfrac{1}{\zeta_k^2}\mathbf{u}_{k-1}.$$

(d) Recall that $\zeta_1 = \frac{\mu}{\rho}$. Since $\mathbf{r}_1 = \mathbf{r}_0 - \frac{1}{\mu}\mathbf{A}\mathbf{r}_0$, we have that

$$\mathbf{r}_1 = \mathbf{r}_0 - \tfrac{1}{\mu}\mathbf{A}\mathbf{u}_0, \quad \mathbf{x}_1 = \mathbf{x}_0 + \tfrac{1}{\mu}\mathbf{u}_0, \quad \text{where} \quad \mathbf{u}_0 = \mathbf{r}_0.$$

(e) Check that this leads to the left algorithm in ALG. 5.5. Compare the costs per step and the memory requirements of this algorithm, with the one in ALG. 5.4.

D Optimal residual polynomials

In LMR, the next residual $\mathbf{r}_{k+1}$ is obtained as an update of $\mathbf{r}_k$ with one vector $\mathbf{c}_k$. With respect to this **residual update vector** $\mathbf{c}_k$, the update is optimal, that is, the updated vector $\mathbf{r}_{k+1}$ has smallest 2-norm. However, residual update vectors $\mathbf{c}_0, \ldots, \mathbf{c}_{k-1}$ from preceding steps have been computed as well and the LMR residual $\mathbf{r}_{k+1}$ need not to have smallest norm with respect to these 'old' update vectors (see Exercise 5.12). The new residual $\mathbf{r}_{k+1}$ can be obtained as the residual of the least square problem $\mathbf{C}_{k+1}\vec{\alpha} = [\mathbf{c}_0, \ldots, \mathbf{c}_k]\vec{\alpha} = \mathbf{r}_k$. For computational reasons (stability), it is convenient to first orthogonalize $\mathbf{c}_k$ against $\mathbf{c}_0, \ldots, \mathbf{c}_{k-1}$. This leads to the generalised conjugate residual (GCR) method (see the left algorithm in ALG. 5.6 and Exercise 5.13).

Since GCR improves on LMR, we have that $\|\mathbf{r}_k^{\mathrm{GCR}}\|_2 \leq \|\mathbf{r}_k^{\mathrm{LMR}}\|_2$ all k (provided that both methods used the same initial guess). In particular, the residual decreases at least with a fixed factor per step if $\frac{1}{2}(\mathbf{A} + \mathbf{A}^*)$ is definite (cf., Exercise 5.6(b)) in which case we can prove that the method does not break down (see Exercise 5.15(c)).

Exercise 5.12. Consider the system

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

(a) Perform a few steps of LMR with $\mathbf{x}_0 = \mathbf{0}$ to solve this system.

(b) What can you tell about the convergence?

(c) Suppose we modify LMR. We update $\mathbf{r}_k$ with the two vector $\mathbf{c}_k$ and $\mathbf{c}_{k-1}$: $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k\mathbf{c}_k + \alpha'_k\mathbf{c}_{k-1}$ and we select the scalars α_k and α'_k such that $\|\mathbf{r}_{k+1}\|_2$ has smallest norm. Apply this modified process to the above system.

Exercise 5.13. **Generalised conjugate residuals.** Consider the left algorithm in ALG. 5.6. To ease discussion, we add an index k to the update of the residual and of the approximate solution: $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k\mathbf{c}_k$ and $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{u}_k$. Moreover, we denote the $\mathbf{u}_k$ and $\mathbf{c}_k$ before the orthogonalization by $\widetilde{\mathbf{u}}_k$ and $\widetilde{\mathbf{c}}_k$, respectively: $\widetilde{\mathbf{u}}_k \equiv \mathbf{r}_k$ and $\widetilde{\mathbf{c}}_k \equiv \mathbf{A}\mathbf{r}_k$.

Assume $\mathbf{c}_k \neq 0$ for all $k = 0, \ldots, m$. Prove the following statements for $k = 0, \ldots, m$.

(a) $\mathbf{c}_k = \mathbf{A}\mathbf{u}_k$, $\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$.

(b) $\mathrm{span}\,(\mathbf{r}_0, \ldots, \mathbf{r}_{k-1}) = \mathrm{span}\,(\mathbf{u}_0, \ldots, \mathbf{u}_{k-1}) = \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$.

(c) $\mathbf{c}_0, \ldots, \mathbf{c}_{k-1}$ forms an orthogonal basis of $\mathbf{A}\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$.

(d) $\mathbf{u}_0, \ldots, \mathbf{u}_{k-1}$ forms an $\mathbf{A}^*\mathbf{A}$-orthogonal basis of $\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$.

(e) $\|\mathbf{r}_k\|_2 \leq \|p_k(\mathbf{A})\mathbf{r}_0\|_2$ for all residual polynomials p_k of degree k.

(f) $\mathbf{r}_k \perp \mathbf{A}\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$.

(g) Assume $\mathbf{A}$ is Hermitian and all eigenvalues are in $[\lambda_-, \lambda_+] \subset (0, \infty)$. Then

$$\|\mathbf{r}_k\|_2 \leq 2\exp\left(-2k\sqrt{\tfrac{\lambda_-}{\lambda_+}}\right)\|\mathbf{r}_0\|_2.$$

Compare this result with those in Exercise 5.5(c) and Exercise 5.10(e). Compare GCR with Richardson as in Exercise 5.5 and with Chebyshev iteration as Exercise 5.10 for Hermitian systems (and for general systems).

Exercise 5.14. **GCR: variants, matrix formulation.**

(a) Give a variant of the GCR-algorithm that relies an orthonormal vectors $\mathbf{c}_j$, i.e., $\|\mathbf{c}_j\|_2 = 1$. Discuss the computational advantages or disadvantages of this variant.

In this exercise, we refer to this 'normalised' variant. Consider the sequence $\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_k$ of GCR residuals. Let $\mathbf{R}_{k+1}$ be the $n \times (k+1)$ matrix with columns of $\mathbf{r}_j$. The matrices $\mathbf{U}_k$ and $\mathbf{C}_k$ are defined similarly.

GCR

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

$\mathbf{x} = \mathbf{x}_0$, $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}_0$

$k = 0$

while $\|\mathbf{r}\|_2 > tol$ do

  $\mathbf{u}_k = \mathbf{r}$

  $\mathbf{c}_k = \mathbf{A}\mathbf{u}_k$

  for $j = 0 : k-1$ do

    $\beta_j = \frac{\mathbf{c}_j^* \mathbf{c}_k}{\sigma_j}$

    $\mathbf{c}_k \leftarrow \mathbf{c}_k - \beta_j \mathbf{c}_j$

    $\mathbf{u}_k \leftarrow \mathbf{u}_k - \beta_j \mathbf{u}_j$

  end for

  $\sigma_k = \mathbf{c}_k^* \mathbf{c}_k$, $\alpha = \frac{\mathbf{c}_k^* \mathbf{r}}{\sigma_k}$

  $\mathbf{r} \leftarrow \mathbf{r} - \alpha\, \mathbf{c}_k$

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

  $k \leftarrow k + 1$

end while

FLEXIBLE GCR

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

$\mathbf{x} = \mathbf{x}_0$, $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}_0$

$k = 0$

while $\|\mathbf{r}\|_2 > tol$ do

  Select $\mathbf{u}_k$ st $\mathbf{A}\mathbf{u}_k \approx \mathbf{r}$

  $\mathbf{c}_k = \mathbf{A}\mathbf{u}_k$

  for $j = 0 : k-1$ do

    $\beta_j = \frac{\mathbf{c}_j^* \mathbf{c}_k}{\sigma_j}$

    $\mathbf{c}_k \leftarrow \mathbf{c}_k - \beta_j \mathbf{c}_j$

    $\mathbf{u}_k \leftarrow \mathbf{u}_k - \beta_j \mathbf{u}_j$

  end for

  $\sigma_k = \mathbf{c}_k^* \mathbf{c}_k$, $\alpha = \frac{\mathbf{c}_k^* \mathbf{r}}{\sigma_k}$

  $\mathbf{r} \leftarrow \mathbf{r} - \alpha\, \mathbf{c}_k$

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

  $k \leftarrow k + 1$

end while

ALGORITHM 5.6. Generalised conjugate residuals (GCR) for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$ with residual accuracy *tol*. The 'for-loop' (here in MATLAB notation) is skipped if $k - 1 < 0$. Note that the scalar β_j is the same in the update of $\mathbf{u}_k$ and of $\mathbf{c}_k$. The version at the left is the standard version (GCR or standard GCR), at the right we have a flexible variant (flexible GCR, cf., Lecture 5.F below).

(b) Let $\mathbf{A}\mathbf{R}_k = \mathbf{Q}_k B_k$ be the QR-decomposition of $\mathbf{A}\mathbf{R}_k$ (in economical form): B_k is $k \times k$ upper triangular. Prove that the columns of $\mathbf{Q}_k$ are equal to the $\mathbf{c}_j$ except for signs (that is, $\mathbf{q}_j = \zeta_j\, \mathbf{c}_j$ for some sign ζ_j, i.e., a scalar $\zeta_j \in \mathbb{C}$ for which $|\zeta_j| = 1$). Assume the signs are all 1, i.e., $\mathbf{A}\mathbf{R}_k = \mathbf{C}_k B_k$

(c) Prove that $\mathbf{U}_k = \mathbf{R}_k B_k^{-1}$ and $\mathbf{r}_{k+1} = (\mathbf{I} - \mathbf{C}_k\, \mathbf{C}_k^*)\mathbf{r}_k = (\mathbf{I} - \mathbf{C}_k\, \mathbf{C}_k^*)\mathbf{r}_0$

Exercise 5.15. **Orthodir.** We apply (unpreconditioned) GCR of ALG. 5.6 to the system $\mathbf{A}\mathbf{x} = \mathbf{b}$ with $\mathbf{A}$ square, non-singular. Here, we denote the $\mathbf{u}$-vectors and $\mathbf{c}$-vectors *before* orthogonalization by $\mathbf{u}'_k$ and $\mathbf{c}'_k$, while $\mathbf{u}_k$ and $\mathbf{c}_k$ are the $\mathbf{u}$-vectors and $\mathbf{c}$-vectors *after* orthogonalization (i.e., $\mathbf{c}'_k \equiv \mathbf{A}\mathbf{u}'_k$ is orthogonalised against $\mathbf{c}_0, \ldots, \mathbf{c}_{k-1}$ to compute $\mathbf{c}_k$).

Note that the $\mathbf{u}'_k$ are used to *expand* the search subspace, whereas the $\mathbf{u}_k$ vectors are used to *extract* the approximate solution from the search subspace, i.e., $\mathbf{x}_k$ is computed as a linear combination of the $\mathbf{u}_j$. Though not discussed here, we note that the fact that the basis for expansion differs form the basis for extraction complicates the effect (and its analysis) of rounding errors on the accuracy.

(a) What happens (depending on the initial guess $\mathbf{x}_0$) when GCR is applied to solve the system

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} ?$$

(b) Discuss the effect of $\mathbf{c}_k^* \mathbf{r}_k = 0$ at step k.

(c) Prove that $\mathbf{c}_k^* \mathbf{r}_k = \mathbf{r}_k^* \mathbf{A}^* \mathbf{r}_k$ and conclude that $\mathbf{c}_k^* \mathbf{r}_k = 0$ does not occur if the the Hermitian part, $\frac{1}{2}(\mathbf{A} + \mathbf{A}^*)$, of $\mathbf{A}$ is positive definite, unless $\mathbf{r}_k = \mathbf{0}$.
(If $\mathbf{A}$ is normal, then the Hermitian part is positive definite if and only if the spectrum of $\mathbf{A}$ is in

the right complex half plane, i.e., $\mathrm{Re}(\lambda) > 0$ for all eigenvalues λ. Why? Does this equivalence also holds for general, non-normal, $\mathbf{A}$?).

(d) Breakdown can be avoided by selecting $\mathbf{u}'_k = \mathbf{c}_{k-1}$ $(k \geq 1)$ for expansion rather than $\mathbf{u}'_k = \mathbf{r}_k$. The resulting method is called **Orthodir** (Orthogonal directions). Explain the naming. Prove that Orthodir does not break down (unless $\mathbf{x}_k = \mathbf{x}$; for ease of discussion, you may assume that $\mathbf{x}_0 = \mathbf{0}$).

(e) Prove that (in exact arithmetic) GCR and Orthodir are equivalent (i.e., they have the same residuals at corresponding steps at the same computational costs) in case GCR does not break down.

(f) Discuss pros and cons of Orthodir versus GCR.

The Krylov subspace $\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$ is the **search subspace** in the kth step of (standard) GCR: the method 'searches' for an approximate solution $\mathbf{x}_{k+1}$ in $\mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$. In step k, the search subspace is expanded by $\widetilde{\mathbf{u}}_k = \mathbf{r}_k$. Before updating the residual and approximate solution, the vector $\widetilde{\mathbf{u}}_k$ is $\mathbf{A}^*\mathbf{A}$-orthogonalised against $\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$.

For each k, GCR computes an approximate solution $\mathbf{x}_k$ in the shifted Krylov subspace $\mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$. In this subspace, $\mathbf{x}_0$ is the best in the sense that it is the vector with residual with smallest norm. Usually matrix-vector multiplications (MVs) are combined with preconditioning. Certainly, in these cases, (preconditioned) MVs are the most expensive computational ingredients of Krylov subspace methods. The other high dimensional operations are vector updates (AXPYs of the form $\alpha\mathbf{x} + \mathbf{y}$) and inner products (DOT of the form $\mathbf{x}^*\mathbf{y}$). When k MVs can be used, then, among all Krylov subspace methods, GCR (and GMRES, to be discussed in the next lecture) is the Krylov subspace method that finds the approximate solution with smallest residual. Nevertheless, this does not necessarily imply that GCR is the fastest method in (computational) time.

> *The number of AXPYs and DOTs that GCR needs per step grows proportionally with the step number.*

If many iteration steps are needed (think of 30 or more), then the computational time for performing AXPYs and DOTs dominates the time for MVs (and even for preconditioned MVs, cf., §Lecture 5.F below).

In following three subsections and the following lectures, we will learn that methods that do not aim to find the 'best' approximation in the shifted Krylov subspace, can be more efficient (in time) by keeping the number of AXPYs and DOTs per step fixed.

E Optimal methods for Hermitian matrices

In the next exercise, we will learn that the number of AXPYs and DOTs per step can be limited if additional information on the algebraic structure of the matrix is known and can be exploited. To be more precise, we will focus on the case where $\mathbf{A}$ is Hermitian, leading to **conjugate residuals** (**CR**), and, where $\mathbf{A}$ is positive definite, leading to **conjugate gradients** (**CG**).

Exercise 5.16. **CR.** Assume $\mathbf{A}$ is $n \times n$ Hermitian. Consider the kth step of GCR: $\mathbf{c}_0, \ldots, \mathbf{c}_{k-1}$ have been constructed (and form an orthogonal basis of $\mathbf{A}\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$, see Exercise 5.13).

(a) Prove that $\mathbf{A}\mathbf{r}_k \perp \mathbf{A}\mathcal{K}_{k-1}(\mathbf{A}, \mathbf{r}_0)$. (Hint: use Exercise 5.13(f).)

(b) Select β such that $\mathbf{A}\mathbf{r}_k - \beta\mathbf{c}_{k-1} \perp \mathbf{c}_{k-1}$. Prove that $\mathbf{A}\mathbf{r}_k - \beta\mathbf{c}_{k-1} \perp \mathbf{A}\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$. Conclude that $\mathbf{c}_0, \ldots, \mathbf{c}_{k-1}, \mathbf{A}\mathbf{r}_k - \beta\mathbf{c}_{k-1}$ form an orthogonal basis of $\mathbf{A}\mathcal{K}_{k+1}(\mathbf{A}, \mathbf{r}_0)$.

(c) Derive the CR algorithm (see ALG. 5.7) and prove that it is mathematically equivalent to GCR (in this case, where $\mathbf{A}$ is Hermitian).

Note that in CR $\mathbf{c} = \mathbf{A}\mathbf{u}$. Replacing the two lines $\widetilde{\mathbf{c}} = \mathbf{A}\widetilde{\mathbf{r}}$ and $\mathbf{c} \leftarrow \widetilde{\mathbf{c}} - \beta\mathbf{c}$ in the CR algorithm of ALG. 5.7 by one line $\mathbf{c} = \mathbf{A}\mathbf{u}$, would save one AXPY. Unfortunately, this

Conjugate Residuals
Select $\mathbf{x}_0 \in \mathbb{C}^n$
$\mathbf{x} = \mathbf{x}_0,\ \mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
$\mathbf{u} = \mathbf{0},\ \mathbf{c} = \mathbf{0},\ \sigma = 1$
while $\|\mathbf{r}\|_2 > tol$ do
$\widetilde{\mathbf{u}} = \mathbf{r},\ \widetilde{\mathbf{c}} = \mathbf{A}\widetilde{\mathbf{u}}$
$\beta = \mathbf{c}^*\widetilde{\mathbf{c}}/\sigma$
$\mathbf{c} \leftarrow \widetilde{\mathbf{c}} - \beta\,\mathbf{c}$
$\mathbf{u} \leftarrow \widetilde{\mathbf{u}} - \beta\,\mathbf{u}$
$\sigma = \mathbf{c}^*\mathbf{c},\ \alpha = \mathbf{c}^*\mathbf{r}/\sigma$
$\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c}$
$\mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u}$
end while

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

Algorithm 5.7. Conjugate residuals (CR) (at the left) and conjugate gradients (CG) (at the right) for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$ with residual accuracy *tol*. $\mathbf{A}$ is assumed to be Hermitian for CR and positive definite for CG.

replacement is not possible since $\widetilde{\mathbf{c}}$ is needed in the computation of β. If $\mathbf{A}$ is positive, we can compute β from $\mathbf{u}$ by switching to the $\mathbf{A}^{-1}$-inner product. This leads to the CG algorithm.

Exercise 5.17. CG. Assume that $\mathbf{A}$ is an $n \times n$ positive definite matrix.

(a) Formulate CR with respect to the $\mathbf{A}^{-1}$ inner product rather than with the standard inner product, i.e., replace expressions as $\mathbf{z}^*\mathbf{y}$ by $\mathbf{z}^*\mathbf{A}^{-1}\mathbf{y}$. Show that the resulting expressions can be evaluated without inverting $\mathbf{A}$.

(b) Note that now, $\mathbf{r}_k$ is the smallest residual with respect to the $\mathbf{A}^{-1}$-inner product, rather than the standard 2-norm. Prove that the residuals form an orthogonal system (orthogonal with respect to the standard inner product), in particular,

$$\mathbf{r}_k \perp \operatorname{span}(\mathbf{r}_0, \ldots, \mathbf{r}_{k-1}) = \operatorname{span}(\mathbf{u}_0, \ldots, \mathbf{u}_{k-1}).$$

(c) Rearrange the lines, to save one AXPY per step.

(d) Note that $\alpha_{k-1}\mathbf{c}_{k-1} = \mathbf{r}_{k-1} - \mathbf{r}_k$ and $\mathbf{u}_{k-1} = \mathbf{r}_{k-1} - \beta_{k-1}\mathbf{u}_{k-2}$. Use the orthogonality relations to prove that

$$\beta_k = \frac{\mathbf{c}_{k-1}^*\mathbf{r}_k}{\mathbf{c}_{k-1}^*\mathbf{u}_{k-1}} = -\frac{\mathbf{r}_k^*\mathbf{r}_k}{\mathbf{r}_{k-1}^*\mathbf{u}_{k-1}} = -\frac{\mathbf{r}_k^*\mathbf{r}_k}{\mathbf{r}_{k-1}^*\mathbf{r}_{k-1}} \quad \text{and} \quad \alpha_k = \frac{\mathbf{u}_k^*\mathbf{r}_k}{\mathbf{c}_k^*\mathbf{u}_k} = \frac{\mathbf{r}_k^*\mathbf{r}_k}{\mathbf{c}_k^*\mathbf{u}_k}.$$

This saves one DOT per step. As a side product, $\|\mathbf{r}_k\|_2^2$ is computed (saving another DOT).

(e) Derive the CG algorithm in Alg. 5.7.

(f) Assume all eigenvalues of $\mathbf{A}$ are in $[\lambda_-, \lambda_+] \subset (0, \infty)$. Prove that

$$\|\mathbf{r}_k\|_{A^{-1}} \le 2\exp\left(-2k\sqrt{\frac{\lambda_-}{\lambda_+}}\right)\|\mathbf{r}_0\|_{A^{-1}} \qquad (k = 0, 1, 2, \ldots).$$

(g) Compare the computational costs and memory requirements of CR, CG and Chebyshev iteration (in the coupled two term variant of Exercise 5.11 and the right algorithm in Alg. 5.5). Discuss pros and cons.

There are several ways to derive CG. Here, we saw CG as a variant of GCR that exploits positive definiteness of the matrix. In following lectures, we will discuss other derivations.

They all lead to the same CG algorithm in case $\mathbf{A}$ is positive definite, but they allow different generalisation (for non-symmetric $\mathbf{A}$ and even for non-linear systems of equations) each with their own pros and cons.

F Nesting and preconditioning

Note that if in LMR the update vector $\mathbf{u}_k$ is selected to be the solution of the system $\mathbf{A}\mathbf{u} = \mathbf{r}_k$ rather than $\mathbf{u} = \mathbf{r}_k$, then $\mathbf{r}_{k+1} = \mathbf{0}$. Similarly, selecting the solution of $\mathbf{A}\widetilde{\mathbf{u}} = \mathbf{r}_k$ for expanding the search subspace in GCR rather than $\widetilde{\mathbf{u}}_k = \mathbf{r}_k$ leads to $\mathbf{r}_{k+1} = \mathbf{0}$ (Why?). Of course solving $\mathbf{A}\widetilde{\mathbf{u}} = \mathbf{r}_k$ is as hard as solving $\mathbf{A}\mathbf{x} = \mathbf{b}$. However, often better approximations of $\mathbf{A}^{-1}\mathbf{r}_k$ than $\widetilde{\mathbf{u}} = \mathbf{r}_k$ are available (or easy to obtain). The **flexible** variant of **GCR** at the right in ALG. 5.6 allows to expand the search subspace by any vector $\widetilde{\mathbf{u}}_k$.

Examples:

- $\widetilde{\mathbf{u}}_k = \mathbf{r}_k$ (**standard GCR**)
- $\widetilde{\mathbf{u}}_k$ solves $\mathbf{M}\widetilde{\mathbf{u}}_k = \mathbf{r}_k$ with $\mathbf{M} \approx \mathbf{A}$ (**preconditioned GCR**)
- $\widetilde{\mathbf{u}}_k$ is the approximate solution of $\mathbf{A}\widetilde{\mathbf{u}} = \mathbf{r}_k$ obtained with s steps of standard GCR (**nested GCR**), where s is the same (fixed) for each k
- $\widetilde{\mathbf{u}}_k$ is an approximate solution of $\mathbf{A}\widetilde{\mathbf{u}} = \mathbf{r}_k$ such that $\|\mathbf{r}_k - \mathbf{A}\widetilde{\mathbf{u}}_k\|_2 \leq 0.1\|\mathbf{r}_k\|_2$.

Of course, if solutions ($\widetilde{\mathbf{x}}$) of "nearby" systems ($\widetilde{\mathbf{A}}\widetilde{\mathbf{x}} = \widetilde{\mathbf{b}}$) are available, then these solutions can also be taken for $\widetilde{\mathbf{u}}_k$ (for the first few k).

Flexible GCR allows to use preconditioners $\mathbf{M}_k$ that vary per step (replace $\mathbf{M}$ by $\mathbf{M}_k$ in the second example). Actually, the third and fourth example can be viewed as examples of varying preconditioners: here $\mathbf{M}_k^{-1} = q_k(\mathbf{A})$ where q is some polynomial (of, in example 3, degree s) that depends on $\mathbf{r}_k$ ($\widetilde{\mathbf{u}}_k = q(\mathbf{A})\widetilde{\mathbf{r}}_k$).

Exercise 5.18. Preconditioned GCR.
Consider the flexible GCR algorithm at the right in ALG. 5.6. Take $\mathbf{x}_0 = \mathbf{0}$.
Let $\mathbf{M}$ be an $n \times n$ matrix (possibly in factorized form) such that

- $\mathbf{M}$ approximates $\mathbf{A}$ in some sense and
- systems $\mathbf{M}\,\widetilde{\mathbf{u}} = \mathbf{r}_k$ are easy to solve,
- $\mathbf{M}$ (or its factors) could "easily" be constructed.

For each k, let, in step k, $\widetilde{\mathbf{u}}_k$ be the solution of the **preconditioner system** $\mathbf{M}\widetilde{\mathbf{u}} = \mathbf{r}_k$. We refer to this variant of flexible GCR as **preconditioned GCR** with **preconditioner** $\mathbf{M}$.

(a) Prove that the resulting residuals $\mathbf{r}_k$ are equal to the residuals of standard GCR applied to the right **preconditioned system** $\mathbf{A}\mathbf{M}^{-1}\mathbf{y} = \mathbf{b}$ (with initial approximate $\mathbf{y}_0 = \mathbf{0}$).

(b) Relate the approximate solutions of these two methods.

(c) Show that preconditioned GCR computes approximate solutions in the Krylov subspace $\mathcal{K}_k(\mathbf{M}^{-1}\mathbf{A}, \mathbf{r}_0)$ with residuals in $\mathcal{K}_{k+1}(\mathbf{A}\mathbf{M}^{-1}, \mathbf{r}_0)$.

The purpose of preconditioning is to reduce the number of MVs that is required to have the solution to certain accuracy (or, equivalently, to have the residual that is in norm sufficiently small). More effective preconditioners will generally better reduce the required number of iteration steps, but they will also increase the computational costs per (preconditioned) MV. Note that by reducing the number of iteration steps, the costly steps (with a high number of AXPYs and DOTs) are somewhat avoided. It is hard to tell in advance what the best (most efficient) procedure will be. It very much depends on the (class of) linear systems that are to be solved. But it is safe to state that iterative methods are efficient only in combination with some preconditioning strategy. We will discuss preconditioners in detail in Lecture 10.A.

In nested GCR (i.e., the third and fourth variant in the above list of examples), there is an **outer loop**, where k is increased and the $\mathbf{u}_k$ and $\mathbf{c}_k$ are being formed, and there are **inner loop**s, where GCR is used to solve $\mathbf{A}\widetilde{\mathbf{u}} = \mathbf{r}_k$ by s steps (third example) or to some residual accuracy (fourth example).

For some classes of (practical relevant) linear problems, it appears that, with nested GCR, the number of MVs that is required to have a certain reduction of the residual norm, is (almost) independent of the number s of GCR steps in the inner loops. Suppose m MVs lead to the required residual reduction. If s is small, then the computational costs in the outer loop are high, while the inner loops are cheap. On the other hand, if s is large (but $s \leq m$), then the inner loops are expensive and the outer loop is cheap. For some intermediate s (as $s = \sqrt{m}$?) the to total computational costs will be minimised.

G Restarts and truncation

In the above derivation of CR and CG, for problems with an $\mathbf{A}$ with some symmetry, mathematical properties have been exploited to find an implementation of GCR that limits memory requirements and computational costs per step. For certain non-symmetric problems, such limitations can also be achieved by nesting. There are more "brute force" approaches that achieve such limitations without relying on symmetry. These approaches give up the idea of finding the best solution in the Krylov search subspace. As a consequence, they usually need (many) more steps. But, nevertheless, since the steps are (relatively) cheap, they can often be very successful. These "brute force" approaches include **restart**, **truncation** and nesting (as mentioned above as example of a flexible GCR variant). In Lecture 8 and Lecture 11.A, we will derive memory friendly and step-wise efficient methods for general matrices based on mathematical arguments. Here (in the exercise below), we will discuss the "brute force" modification of GCR.

To avoid confusion, in the context of restarted GCR and truncated GCR, we also refer to GCR as **full GCR**.

Consider the GCR algorithm of Alg. 5.6. Let ℓ be a positive integer.

We denote the version of GCR that is *restarted* every ℓth step by GCR(ℓ): starting with $\mathbf{x}_0$ and $\mathbf{r}_0$, after ℓ steps, we have computed $\mathbf{x}_\ell$ and $\mathbf{r}_\ell$. We then restart by taking $\mathbf{x}_\ell$ as initial guess for a new cycle of ℓ GCR steps ($\mathbf{x}_0 \leftarrow \mathbf{x}_\ell$, and, therefore, $\mathbf{r}_0 \leftarrow \mathbf{r}_\ell$) and repeat this restart procedure until we have sufficient accuracy.

The version of GCR that *truncates* the orthogonalisation procedure to the last ℓ vectors is denoted by ℓ-GCR, that is, $\mathbf{c}_k$ is obtained from $\mathbf{c}'_k$ ($\equiv \mathbf{A}\mathbf{u}'_k$ with $\mathbf{u}'_k \equiv \mathbf{r}_k$) by orthogonalising $\mathbf{c}'_k$ only against the ℓ preceding vectors $\mathbf{c}_{k-1}, \ldots, \mathbf{c}_{k-\ell}$ (if $k - \ell \geq 0$). Thus forming an orthogonal system $\mathbf{c}_k, \mathbf{c}_{k-1}, \ldots, \mathbf{c}_{k-\ell}$ of $\ell + 1$ vectors. Note that the $\mathbf{c}_k$ of ℓ-GCR will generally not be orthogonal to, say, $\mathbf{c}_0$. In particular, this vector $\mathbf{c}_k$ will be different from the $\mathbf{c}_k$ of GCR. The $\mathbf{u}_k$ in ℓ-GCR is obtained accordingly from $\mathbf{u}'_k$ and $\mathbf{u}_{k-1}, \ldots, \mathbf{u}_{k-\ell}$.

Exercise 5.19.

(a) Show that both GCR(ℓ) and ℓ-GCR can be obtained by replacing the line

`for` $j = 0 : k - 1$ `do`

in GCR of Alg. 5.6 by the line

`for` $j = \pi(k) : k - 1$ `do`,

where, in case of GCR(ℓ), $\pi(k) \equiv m(\ell + 1)$ with $m \in \mathbb{N}_0$ maximal such that $k \geq m(\ell + 1)$, and $\pi(k) = \max(0, k - \ell)$ in case of ℓ-GCR.

(b) Analyse the storage that is required in both GCR variants.
Analyse the (average) computational costs (average with respect to the number of MVs).

(c) Show that both 0-CGR and GCR(1) coincide with LMR. Show that 1-GCR equals CR. Generalize Exercise 5.6 and show that GCR(ℓ) converges for each $\ell \geq 1$ if $\frac{1}{2}(\mathbf{A} + \mathbf{A}^*)$ is definite.

In case $\mathbf{A}$ is Hermitian, 1-GCR takes as many steps (MVs) as GCR. However, for non-Hermitian $\mathbf{A}$, 1-GCR need not to converge even in cases where $\frac{1}{2}(\mathbf{A} + \mathbf{A}^*)$ is definite and LMR converges quickly. Usually, GCR(ℓ) needs less steps (less MVs) to converge than GCR(ℓ') if $\ell > \ell'$. However, there are examples where GCR(2) needs less steps than GCR(ℓ) for $\ell > 2$. Try to illustrate the above observations with numerical examples.

The following two exercises are included for referential purposes: the results will be used in Lecture 8 and Lecture 11.A to derive some iterative solvers for linear systems.

Exercise 5.20. **Truncated GCR.** We now consider ℓ-GCR.

(a) Show that for all k, and for some scalars α_k and $\eta_k^{(j)}$ $(j = 1, \ldots, \ell)$, we have that

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{r}_k - \eta_k^{(1)}(\mathbf{r}_k - \mathbf{r}_{k-1}) - \ldots - \eta_k^{(\ell)}(\mathbf{r}_{k+1-\ell} - \mathbf{r}_{k-\ell}).$$

(Hint: $\mathbf{c}_k = \mathbf{A}\mathbf{r}_k - \beta_{k-1}\mathbf{c}_{k-1}$, $\mathbf{r}_{j+1} = \mathbf{r}_j - \alpha_j \mathbf{c}_j$).

(b) Show that these scalars α_k and $\eta_k^{(j)}$ solve

$$\min_{\alpha, \eta^{(1)}, \ldots, \eta^{(\ell)}} \|\mathbf{r}_k - \alpha \mathbf{A}\mathbf{r}_k - \eta^{(1)}(\mathbf{r}_k - \mathbf{r}_{k-1}) - \ldots - \eta^{(\ell)}(\mathbf{r}_{k+1-\ell} - \mathbf{r}_{k-\ell})\|_2,$$

i.e., the $(\ell+1)$-vector $\vec{\alpha}_k \equiv (\alpha_k, \eta_k^{(1)}, \ldots, \eta_k^{(\ell)})^{\mathrm{T}}$ solves the equation $[\mathbf{A}\mathbf{r}_k, \Delta_k^r]\,\vec{\alpha}_k = \mathbf{r}_k$ in the least square sense. Here, $\Delta_k^r \equiv [\,\mathbf{r}_k - \mathbf{r}_{k-1}, \ldots, \mathbf{r}_{k+1-\ell} - \mathbf{r}_{k-\ell}\,]$ is the $\ell \times n$ matrix of residual differences. In particular, the vector $\vec{\alpha}_k$ can be computed by solving the associated normal equations. Conclude that $\vec{\alpha}_k$ is real if all entries of $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{x}_0$ are real.

(c) Show that the $\ell \times n$ matrix Δ_k^r of residual differences is orthogonal.

(d) Show that for the above least square solution $\vec{\alpha}_k$, we have

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{r}_k + \eta_k^{(1)}(\mathbf{x}_k - \mathbf{x}_{k-1}) + \ldots + \eta_k^{(\ell)}(\mathbf{x}_{k+1-\ell} - \mathbf{x}_{k-\ell}) = \mathbf{x}_k + [\mathbf{r}_k, \Delta_k^u]\,\vec{\alpha}_k.$$

Here, Δ_k^u denotes the $\ell \times n$ matrix of approximate solution differences.

(e) Now take $\ell = 1$. For each k, there is a polynomial p_k of exact degree k with $p_k(0) = 1$ such that $\mathbf{r}_k = p_k(\mathbf{A})\mathbf{r}_0$. Show that the (p_k) satisfy the polynomial three-term recurrence

$$p_{k+1}(\zeta) = (1 - \alpha_k \zeta)p_k(\zeta) + \eta_k(p_k(\zeta) - p_{k-1}(\zeta)) \qquad (\zeta \in \mathbb{C}).$$

Prove that p_k and p_{k-1} do not share a zero (if they have a common zero, then so do p_{k-1} and p_{k-2}, etc.).

Exercise 5.21. **Restarted GCR.** Consider GCR(ℓ). Let m be a multiple of ℓ: $m = j\ell$ and $k = m + \ell = (j+1)\ell$.

(a) Show that for some scalars $\beta_m^{(1)}, \ldots, \beta_m^{(\ell)}$,

$$\mathbf{r}_k = \mathbf{r}_m - \beta_m^{(1)}\mathbf{A}\mathbf{r}_m - \beta_m^{(2)}\mathbf{A}^2\mathbf{r}_m - \ldots - \beta_m^{(\ell)}\mathbf{A}^\ell\mathbf{r}_m.$$

(b) Show that these scalars

$$\min_{\beta^{(1)}, \ldots, \beta^{(\ell)}} \|\mathbf{r}_m - \beta^{(1)}\mathbf{A}\mathbf{r}_m - \ldots - \beta^{(\ell)}\mathbf{A}^\ell\mathbf{r}_m\|_2,$$

i.e., the ℓ-vector $\vec{\beta}_m \equiv (\beta_m^{(1)}, \beta_m^{(2)}, \ldots, \beta_m^{(\ell)})^{\mathrm{T}}$ solves the equation $\mathbf{A}\mathbf{R}_m\,\vec{\beta}_m = \mathbf{r}_m$ in the least square sense. Here, $\mathbf{R}_m$ is the $\ell \times n$ matrix $\mathbf{R}_m \equiv [\,\mathbf{r}_m, \mathbf{A}\mathbf{r}_m, \ldots, \mathbf{A}^{\ell-1}\mathbf{r}_m\,]$. In particular, the vector $\vec{\beta}_m$ can be computed by solving the associated normal equations.
Show that $\vec{\beta}_m$ is real if all entries of $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{x}_0$ are real.

(c) Show that

$$\mathbf{x}_k = \mathbf{x}_m + \beta_m^{(1)}\mathbf{r}_m + \beta_m^{(2)}\mathbf{A}\mathbf{r}_m + \ldots + \beta_m^{(\ell-1)}\mathbf{A}^\ell\mathbf{r}_m = \mathbf{x}_m + \mathbf{R}_m\,\vec{\beta}_m.$$

(d) There are polynomials p_k such that $\mathbf{r}_k = p_k(\mathbf{A})\mathbf{r}_0$. For $m = j\ell$, put

$$q_j(\zeta) \equiv 1 - \beta_m^{(1)}\zeta - \ldots - \beta_m^{(\ell)}\zeta^\ell \qquad (\zeta \in \mathbb{C}).$$

Show that, $p_\ell = q_0$ and

$$p_{m+\ell}(\zeta) = q_j(\zeta)\,p_m(\zeta) = q_j(\zeta)\,q_{j-1}(\zeta) \cdot \ldots \cdot q_0(\zeta) \qquad (\zeta \in \mathbb{C}):$$

p_m is a product of j residual polynomials of degree ℓ.
Conclude that any zero of p_m is a zero of $p_{m+\ell}$ as well.
Conclude that GCR(2) and 1-GCR can not be the same (i.e., not all kth residuals can be the same).

(e) Any polynomial of degree ℓ can be factorized as a product of ℓ polynomial factors of degree 1 (main theorem of the algebra). Show that q_j is a real polynomial (i.e., all its coefficients are real) if all entries of $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{x}_0$ are real. Is that also the case for the degree 1 factors of q_j? Conclude that, for $\ell \geq 2$ GCR(1) will generally not be the same as GCR(ℓ) (i.e., not all kth residuals can be the same for $k = j\ell$).

H Krylov subspaces and Hessenberg matrices

Let $\mathbf{A}$ be an $n \times n$ matrix. Let $\mathbf{b}$ a non-trivial n-vector.

Note that the order of the Krylov subspace equals the number of generating vectors. The order need not be equal to the dimension:

Exercise 5.22. Let m be the largest number for which $\mathcal{K}_{m-1}(\mathbf{A}, \mathbf{b}) \neq \mathcal{K}_m(\mathbf{A}, \mathbf{b})$.

(a) Prove that $m \leq n$.

(b) Show that $\mathcal{K}_k(\mathbf{A}, \mathbf{b}) = \mathcal{K}_\ell(\mathbf{A}, \mathbf{b})$ for all $k \geq m$.

(c) Show that the order k of $\mathcal{K}_k(\mathbf{A}, \mathbf{b})$ equals the dimension $\mathcal{K}_k(\mathbf{A}, \mathbf{b}) \quad \Leftrightarrow \quad k \leq m$.

The following theorem tells us that we may assume that all eigenvalues are simple (i.e., $\mathbf{A}$ is simple) if $\mathbf{A}$ is diagonalizable (semi-simple) and we work with a Krylov subspace method. More general, we may assume that in the Jordan normal form of $\mathbf{A}$, different Jordan blocks have different eigenvalues (see Exercise 5.31).

Theorem 5.3 *Assume* $\mathbf{A}$ *is diagonalizable.*
Then all eigenvalues of the restriction of $\mathbf{A}$ *to* $\mathcal{K}(\mathbf{A}, \mathbf{b}) \equiv \text{span}\{\mathbf{A}^k\mathbf{b} \,\big|\, k = 0, 1, 2, \ldots\}$ *are simple. In particular, the dimension of the Krylov subspace of order* k *is less than* k *and less than or equal to the number of different eigenvalues of* $\mathbf{A}$.

Exercise 5.23. ***Proof of Theorem*** 5.3**.** Let $\mathbf{A}$ be an $n \times n$ diagonalizable matrix and let $\mathbf{b}$ be an n-vector.

It is convenient for this exercise to express $\mathbf{b}$ as a linear combination

$$\mathbf{b} = \mathbf{v}_1 + \ldots + \mathbf{v}_m$$

of eigenvectors $\mathbf{v}_j$ with mutually different eigenvalues, i.e., $\mathbf{A}\mathbf{v}_j = \lambda_j \mathbf{v}_j$ and $\lambda_i \neq \lambda_j$ for all $i, j = 1, \ldots, m$, $i \neq j$.

(a) Show that such a decomposition exists in case $\mathbf{A}$ is the 2×2 identity matrix. What is the value for m in this case?

Show that such a decomposition exists if $\mathbf{A}$ is diagonalizable. Note that the eigenvectors depend on $\mathbf{b}$.

(b) Let $\mathcal{V}$ be the span of $\mathbf{v}_1, \ldots, \mathbf{v}_m$. Show that $\mathbf{A}$ maps $\mathcal{V}$ into $\mathcal{V}$: the space $\mathcal{V}$ is **invariant** under multiplication by $\mathbf{A}$. In particular, we have that the Krylov subspace

$$\mathcal{K}_k(\mathbf{A}, \mathbf{b}) \equiv \text{span}(\mathbf{b}, \mathbf{A}\mathbf{b}, \ldots, \mathbf{A}^{k-1}\mathbf{b})$$

of order k is a subspace of $\mathcal{V}$. Show that $\mathbf{v}_1, \ldots, \mathbf{v}_m$ form a basis of $\mathcal{V}$.

(c) Consider the **Vandermonde matrix**

$$V = \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \ldots & \lambda_1^{k-1} \\ 1 & \lambda_2 & \lambda_2^2 & \ldots & \lambda_2^{k-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \lambda_m & \lambda_m^2 & \ldots & \lambda_m^{k-1} \end{bmatrix}$$

Show that the columns of V represent basis vectors of $\mathcal{K}_k(\mathbf{A}, \mathbf{b})$ with respect to the $\mathbf{v}_i$.

Note that, if p is the polynomial $p(x) = \alpha_0 + \alpha_1 x + \ldots + \alpha_{k-1}x^{k-1}$, and $\vec{\alpha}$ is the vector $\vec{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_{k-1})^{\mathrm{T}}$, then $V\vec{\alpha}$ is the vector with coordinates $p(\lambda_j)$. Since p is of degree $< k$, this implies that V is of full rank if $k \leq m$ (Why?).

(d) Conclude that the order k of the Krylov subspace $\mathcal{K}_k(\mathbf{A}, \mathbf{b})$ equals its dimension if and only if $k \leq m$, where m is number of eigenvector components of $\mathbf{b}$ corresponding to different eigenvalues.

Exercise 5.24. Let m be the maximal order for which the dimension of $\mathcal{K}_m(\mathbf{A}, \mathbf{b})$ equals m (see Exercise 5.22).

Assume $\mathbf{A}$ is on Jordan normal form (cf., Theorem 0.7) and assume that all eigenvalues are the same: $\mathbf{A}$ consists of several Jordan blocks J_λ (with J_λ as in Theorem 0.7) possibly of different size, but with the same λ. Let ℓ be the size of the largest Jordan block, i.e., the largest J_λ is $\ell \times \ell$.

(a) Prove that $m \leq \ell$.

(b) Show that the Jordan normal form of $\mathbf{A}$ restricted to $\mathcal{K}_m(\mathbf{A}, \mathbf{b})$ consists of exactly one Jordan block J_λ of size m.

As a combination of the above result with the one in Theorem 5.3 suggests, we have the following result for a general matrix $\mathbf{A}$.

Property 5.4 *Let m be the maximal order for which the dimension of $\mathcal{K}_m(\mathbf{A}, \mathbf{b})$ equals m. Let, for each eigenvalue λ of $\mathbf{A}$, $\ell(\lambda)$ be the size of the largest Jordan block J_λ in the Jordan normal form of $\mathbf{A}$. Then, $m \leq \sum \ell(\lambda)$, where we sum over all* different *eigenvalues λ of $\mathbf{A}$.*

Krylov subspaces and Hessenberg matrices are closely related.

Theorem 3.4 tells us that it make sense to try to construct a partial Hessenberg decomposition, as in the Arnoldi process.

Below $\mathbf{A}$ is an $n \times n$ matrix and $\mathbf{v}_0, \ldots, \mathbf{v}_k$ is a set of $k+1$ linearly independent n-vectors. We put $\mathbf{V}_j \equiv [\mathbf{v}_0, \ldots, \mathbf{v}_{j-1}]$ and $\mathcal{V}_j \equiv \mathrm{span}(\mathbf{V}_j) \quad (j \leq k)$.

Theorem 5.5 *The following three properties are equivalent.*

1) $\mathbf{A}(\mathcal{V}_k) \subset \mathrm{span}\,(\mathcal{V}_k) \oplus [\mathbf{v}_k]$.
2) *There is a $(k+1) \times k$ matrix $\underline{H}_k$ such that $\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\,\underline{H}_k$.*
3) *$\mathcal{V}_{k+1}$ is Krylov subspace or order $k+1$.*

The set $\mathbf{v}_0, \ldots, \mathbf{v}_k$ is said to be a **Krylov basis** or **Krylov flag** if

$$\mathcal{K}_{j+1}(\mathbf{A}, \mathbf{v}_0) = \mathrm{span}(\mathbf{v}_0, \ldots, \mathbf{v}_j) \qquad \text{all} \quad j \leq k. \tag{5.6}$$

An Hessenberg matrix $\underline{H}_k$ is **unreduced** if $H_{i+1,i} \neq 0$ all i.[1]

Theorem 5.6 *Assume $\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\,\underline{H}_k$. Then,*

$$\underline{H}_k \text{ is unreduced Hessenberg} \quad \Leftrightarrow \quad \mathbf{v}_0, \ldots, \mathbf{v}_k \text{ is a Krylov basis.} \tag{5.7}$$

Theorem 5.7 *Assume $\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\,\underline{H}_k$, $\mathbf{v}_0, \ldots, \mathbf{v}_k$ is a Krylov basis and $\mathbf{A}$ is non-singular. Then, $\underline{H}_k$ has full rank and*

$$\mathbf{v}_j \text{ are residuals } (j \leq k) \quad \Leftrightarrow \quad \mathbf{1}^*\underline{H}_k = \mathbf{0}^*. \tag{5.8}$$

[1] The naming 'unreduced' comes from the theory for the QR-algorithm, where the purpose is to 'reduce' an upper Hessenberg matrix to an upper triangular matrix (Schur form). A 0 on the first lower co-diagonal means a step towards the reduction of the matrix to upper triangular form: the QR-algorithm can be reduced to (two) processes on lower dimensional matrices.

Exercise 5.25. ***Proof of Theorem*** 5.6**.** Assume $\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\,\underline{H}_k$. Prove (5.7).

Exercise 5.26. ***Proof of Theorem*** 5.7**.** Assume $\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\,\underline{H}_k$, and $\mathbf{v}_0, \ldots, \mathbf{v}_k$ is a Krylov basis, or, equivalently, $\underline{H}_k$ is unreduced Hessenberg.

(a) Prove that $\underline{H}_k$ has full rank if $\mathbf{A}$ is non-singular.

(b) For each $j \leq k$, there is a polynomial p_j of exact degree j such that $\mathbf{v}_j = p_j(\mathbf{A})\mathbf{v}_0$. Prove this and show that

$$\zeta\,[p_0(\zeta), p_1(\zeta), \ldots, p_{k-1}(\zeta)] = [p_0(\zeta), p_1(\zeta), \ldots, p_k(\zeta)]\,\underline{H}_k \qquad (\zeta \in \mathbb{C}).$$

Note that $[p_0(\zeta), p_1(\zeta), \ldots, p_k(\zeta)]$ is a row vector.

(c) Put $\vec{\gamma}_k \equiv [p_0(0), p_1(0), \ldots, p_k(0)]^*$. Show that

$$e_1^*\vec{\gamma}_k = 1 \quad \text{and} \quad \vec{\gamma}_k^*\,\underline{H}_k = \mathbf{0}^*. \tag{5.9}$$

Prove that (5.9) determines $\vec{\gamma}_k$ for any $(k+1) \times k$ unreduced Hessenberg matrix.

(d) Since $\mathbf{v}_j$ is a residual (i.e., $\mathbf{v}_j = \mathbf{v}_0 - \mathbf{A}\mathbf{x}_j$ for some $\mathbf{x}_j$ is $\mathcal{K}_{j-1}(\mathbf{A}, \mathbf{v}_0)$) if and only if $p_j(0) = 1$ (then $\mathbf{x}_j = q_j(\mathbf{A})\mathbf{v}_0$ with q_j such that $p_j(\zeta) = 1 - \zeta q_j(\zeta)$), conclude that (5.8) holds.

(e) Let $\underline{H}_k = \underline{J}U$ be the LU-decomposition of $\underline{H}_k$ with $\underline{J}$ a lower triangular $(k+1) \times k$ matrix with 1 on the diagonal and U upper triangular $k \times k$ matrix. Show that $\underline{J}$ is Hessenberg and bi-diagonal. Show that

$$\mathbf{1}^*\,\underline{H}_k = \mathbf{0}^* \quad \Leftrightarrow \quad \text{the lower diagonal of } \underline{J} \text{ consists of } -1.$$

(f) Let $\vec{\gamma}_k$ be as in (5.9). Put $D_{k+1} \equiv \mathrm{diag}(\vec{\gamma}_k)$. Show that $\mathbf{1}^*D_{k+1}\underline{H}_k = \mathbf{0}^* = \mathbf{1}^*D_{k+1}\underline{H}_k D_k^{-1}$ and conclude that $\frac{1}{e_j^*\vec{\gamma}_k}\mathbf{v}_j$ (i.e., the columns of $\mathbf{V}_{k+1}D_{k+1}^{-1}$) are residuals.
In case $\mathbf{v}_0, \ldots, \mathbf{v}_k$ is an orthonormal system (as in the Arnoldi relation), the $\mathbf{v}_j$ scaled in the indicated way, are the residuals of th FOM-process.

The following exercise proves Theorem 5.5. This exercise also gives an explicit construction of the Krylov basis of $\mathcal{V}_{k+1}$ (using Householder reflections).

Exercise 5.27. ***Proof of Theorem*** 5.5**.** For ease of notation, we adopt the following conventions in this exercise. If v is an ℓ-vector and we use the vector in an m-dimensional setting, with $m > \ell$, then we assume that v has been expanded with 0's to an m-vector $((v^{\mathrm{T}}, 0, \ldots, 0)^{\mathrm{T}})$. If an $\ell \times \ell$ matrix A is used in an m-dimensional context, then we assume that A has been expanded with zeros to an $m \times m$ matrix, except on the new diagonal entries which are equal to 1.

Let $\underline{G}$ be an $(\ell+1) \times \ell$ matrix.

(a) Let g^{T} be the $(\ell+1)$th row of of $\underline{G}$. Construct a $\|\cdot\|_2$-normalised ℓ-vector v such that the Householder reflection $H_1 \equiv I_\ell - 2\mathbf{v}\mathbf{v}^*$ maps g to a multiple of the ℓth standard basis vector e_ℓ. Let $\underline{G}_1 \equiv H_1\underline{G}H_1$. What is the form of the last row of this matrix?.

(b) Let $\underline{G}^{(1)}$ the $\ell \times (\ell-1)$ left upper block of $\underline{G}_1$. Apply the procedure from (a) to $\underline{G}^{(1)}$ to form an $\ell \times (\ell-1)$ with last row a multiple of $e_{\ell-1}^{\mathrm{T}}$.

(c) Repeat this procedure $\ell - 1$ times and conclude that there is an $\ell \times \ell$ unitary matrices Q ($Q = H_1 \cdot H_2 \cdot \ldots \cdot H_{\ell-1}$) such that $\underline{H} \equiv Q^*\underline{G}Q$ is $(\ell+1) \times \ell$ upper Hessenberg: there is an **Hessenberg decomposition**

$$\underline{G} = Q\underline{H}Q^*, \tag{5.10}$$

with Q unitary and $\underline{H}$ upper Hessenberg.

(d) Assume that $\mathbf{A}\mathbf{W}_\ell = \mathbf{W}_{\ell+1}\underline{G}$ for some full rank $n \times (\ell+1)$ matrix $\mathbf{W}_{\ell+1} = [\mathbf{W}_\ell, \mathbf{w}_\ell] = [\mathbf{w}_0, \ldots, \mathbf{w}_\ell]$. Show that $\mathbf{A}\mathbf{V}_\ell = \mathbf{V}_{\ell+1}\underline{H}$ and $\mathrm{span}\,(\mathbf{V}_{\ell+1}) = \mathrm{span}\,(\mathbf{W}_{\ell+1})$. Here, $\mathbf{V}_{\ell+1} \equiv \mathbf{W}_{\ell+1}Q$. (Pay attention to the dimensions of the matrices).

(e) Prove Theorem 5.5.

Exercise 5.28. Let H be a $k \times k$ unreduced Hessenberg matrix.

(a) Prove or disprove (give a counter example the following claims:

- H is irreducible (see Footnote 5).
- H has full rank.
- If H is irreducible, then H has full rank.
- If H has full rank, then H is irreducible.

(Hint: for 3×3 counter examples, you may assume that H is of the form $H = \begin{bmatrix} * & * & * \\ 1 & * & * \\ 0 & 1 & * \end{bmatrix}$.)

(b) Prove: H is non-singular $\Leftrightarrow$ $\vec{\gamma}^* H = \vec{0}^*$ for some vector $\vec{\gamma}$ with first coordinate 1.

(c) Prove: $\mathbf{e}_1 \in \mathcal{R}(H)$ $\Leftrightarrow$ H is non-singular (Hint: see Exercise 0.40.2).

(d) Assume the unreduced Hessenberg matrix H is tri-diagonal. Let H_m be the $m \times m$ left upper block of H $(m \leq k)$. Prove: if H_{m-1} is singular, then H_m is non-singular.
(Note that the result follows from Exercise 4.15(b) in case H is Hermitian.)

Exercise 5.29. **Hessenberg matrices and the power method.** Let $\mathbf{H} = (H_{i,j})$ be an $n \times n$ upper Hessenberg matrix, i.e., except for the first co-diagonal $\mathbf{H}$ has a zero strict lower diagonal part ($H_{ij} = 0$ if $i > j + 1$). We assume $\mathbf{H}$ to be diagonalizable.

Suppose there is a $j \in \{1, \ldots, n-1\}$ such that also $H_{j+1,j} = 0$. Partition $\mathbf{H}$ as

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 & \mathbf{E} \\ \mathbf{0} & \mathbf{H}_2 \end{bmatrix},$$

where $\mathbf{H}_1$ is the left upper $j \times j$ block and $\mathbf{H}_2$ is the bottom right $(n-j) \times (n-j)$ block. Partition vectors accordingly: $\mathbf{x} = (\mathbf{x}_1^{\mathrm{T}}, \mathbf{x}_2^{\mathrm{T}})^{\mathrm{T}}$.

(a) Show that the set $\Lambda(\mathbf{H})$ of all eigenvalues of $\mathbf{H}$ is the union of $\Lambda(\mathbf{H}_1)$ and $\Lambda(\mathbf{H}_2)$.

(b) Express the eigenvectors of $\mathbf{H}$ in terms of eigenvectors of $\mathbf{H}_1$ and $\mathbf{H}_2$.

We now assume $\mathbf{H}$ to be unreduced. of order k is of dimension $\min(k, n)$.

(c) Conclude that $\mathbf{e}_1$ has a component in the direction of each eigenvector of $\mathbf{H}$. Conclude that the power method started with $\mathbf{x}_0 = \mathbf{e}_1$ will converge to the dominant eigenvector if $\mathbf{H}$ has a dominant eigenvector.

I Unreduced Hessenberg matrices and minimal polynomial

Let H be a $k \times k$ upper Hessenberg matrix.
For each eigenvalue ϑ of H, let $\mu(\vartheta)$ be the multiplicity of the eigenvalue.

Proposition 5.8 *H is unreduced if and only if* $\dim(\mathcal{K}_k(H, e_1)) = k$.

Exercise 5.30. Prove Prop. 5.8.

Theorem 5.9 *Let H be unreduced. Let p be a polynomial of degree k.*
The following three properties are equivalent

1) $p(H)e_1 = 0$.

2) $p(H) = 0$.

3) $p^{(j)}(\vartheta) = 0$ *for all eigenvalues ϑ of H and all $j < \mu(\vartheta)$.*

If p is monic of exact degree k, then we have that

$$p(H)e_1 = 0 \quad \Leftrightarrow \quad p(H) = 0 \quad \Leftrightarrow \quad p \text{ is the minimal polynomial.}$$

Exercise 5.31. ***Proof of Theorem*** 5.9. Let $p(\zeta) = \alpha_0 + \alpha_1\zeta + \ldots + \alpha_k\zeta^k$ ($\zeta \in \mathbb{C}$) be a polynomial. Let $\mathbf{A}$ be a square matrix. The matrix $p(\mathbf{A})$ is defined by

$$p(\mathbf{A}) \equiv \alpha_0\mathbf{I} + \alpha_1\mathbf{A} + \ldots + \alpha_k\mathbf{A}^k$$

For an eigenvalue λ of $\mathbf{A}$, let $\ell(\lambda)$ be the size of the largest Jordan block associated with the eigenvalue λ.

(a) Prove that $p(\mathbf{A}) = \mathbf{0}$ if and only if $p^{(j)}(\lambda) = 0$ for all eigenvalues λ of $\mathbf{A}$ and all $j < \ell(\lambda)$: cf., Theorem 0.9 and Exercise 0.19.

Consider a k by k unreduced upper Hessenberg matrix $H = (H_{i,j})$.

(b) Prove that the degree of the minimal polynomial Q_H of H is k. In particular, $P_H = Q_H$, where P_H is the characteristic polynomial of H.

(c) Show that $p = q$, if both p and q are monic polynomials of exact degree k with $p(H)e_1 = q(H)e_1 = 0$.

(d) Prove Theorem 5.9.

(e) Are the eigenvalues of an unreduced Hessenberg matrix simple? (Hint: consider (a rotated version of) the 2×2 matrix of all 1, except for the $(1, 2)$ entry which equals 0).

(f) Show that if an unreduced Hessenberg matrix is transformed to Jordan normal form, then there are no two Jordan blocks with the same eigenvalue.

J Unreduced Hessenberg matrices and eigenpairs

Leslie matrices are Hessenberg matrices of special type: they have only non-zeros on the first row and the first lower co-diagonal. In addition, these entries are non-negative, but the sign is irrelevant for the following statement. For Leslie matrices, eigenvectors can easily be expressed in terms of the eigenvalues. For instance, with $\beta_1 \equiv 1$, $\beta_j \equiv h_1 \cdot \ldots \cdot h_{j-1} = \beta_{j-1}\, h_{j-1}$ ($j = 2, \ldots, k$), it can easily be checked that

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_{k-1} & \alpha_k \\ h_1 & 0 & \ldots & 0 & 0 \\ 0 & h_2 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & & \vdots \\ 0 & & & h_{k-1} & 0 \end{bmatrix} \begin{bmatrix} \lambda^{k-1} \\ h_1\lambda^{k-1} \\ \beta_3\lambda^{k-3} \\ \vdots \\ \beta_k \end{bmatrix} = \lambda \begin{bmatrix} \lambda^{k-1} \\ h_1\lambda^{k-1} \\ \beta_3\lambda^{k-3} \\ \vdots \\ \beta_k \end{bmatrix}, \tag{5.11}$$

whenever λ is such that $\alpha_1\lambda^{k-1} + \alpha_2\beta_2\lambda^{k-2} + \ldots + \alpha_k\beta_k = \lambda^k$ (check this). This last equation is equivalent to the characteristic polynomial equation $\det(\lambda I - H) = 0$.

The following result generalises this result to general unreduced Hessenberg matrices.

Exercise 5.32. Let $H_k = (h_{ij})$ be an $k \times k$ unreduced upper Hessenberg matrix.
Put $\beta_1 \equiv 1$, $\beta_j \equiv h_{21} \cdot \ldots \cdot h_{j,j-1}$, let H_j be the left $j \times j$ upper block of H_k ($j = 2, \ldots, k$).

(a) Show there is a unique k-vector $\vec{\gamma}_k = (1, \gamma_2, \gamma_3, \ldots, \gamma_k)^{\mathrm{T}}$ such that

$$\vec{\gamma}_k^*\, H_k = \tau_k\, e_k^* \quad \text{for some scalar } \tau_k.$$

Note that $\vec{\gamma}_k$ is scaled to have first coordinate equal to 1.

To find an expression for τ_k and the γ_j, consider the LU-factorisation $H = LU$ of H.

(b) Show that L is of the form

$$L = \begin{bmatrix} 1 & 0 & \ldots & 0 & 0 \\ -\mu_2 & 1 & \ldots & & 0 \\ 0 & -\mu_3 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & & \vdots \\ 0 & & & -\mu_k & 1 \end{bmatrix}.$$

Let u_{ii} be the ith diagonal entry of the upper triangular matrix U.
Show that $h_{j,j-1} = -\mu_j\, u_{j-1,j-1}$ and $u_{11} \cdot \ldots \cdot u_{jj} = \det(H_j)$ $(j = 2, \ldots, k)$.
(c) Show that $\gamma_j = 1/(\mu_2 \cdot \ldots \cdot \mu_j)$ for $j = 2, \ldots, k$ and $\tau_k = \gamma_k u_{kk}$.
(d) Show that

$$\gamma_j = (-1)^{j-1}\, \frac{\det(H_{j-1})}{\beta_j} = \frac{\det(-H_{j-1})}{\beta_j} \quad (j = 2, \ldots, k) \quad \text{and} \quad \tau_k = -\frac{\det(-H_k)}{\beta_k}.$$

(e) For $\lambda \in \mathbb{C}$, let $\vec{\gamma}_k(\lambda)$ be the k-vector with first coordinate 1 and other coordinates given by

$$\gamma_j(\lambda) \equiv \frac{\det(\lambda\, I_{j-1} - H_{j-1})}{\beta_j} \qquad (j = 2, \ldots, k).$$

Show that

$$\vec{\gamma}_k(\lambda)^*\, H_k = \lambda\, \vec{\gamma}_k(\lambda)^* - \frac{\det(\lambda\, I_k - H_k)}{\beta_k}\, e_k^*$$

Conclude that for each eigenvalue ϑ_j of H_k, the vector $\vec{\gamma}_k(\vartheta_j)$ forms a left eigenvector of H_k.
(f) The permutation matrix J that renumbers the coordinates backwards, $Je_1 = e_k$, $Je_2 = e_{k-1}$, etc., can be used to express right eigenvectors of H_k as left eigenvectors of a related upper Hessenberg matrix. Prove that

$$(H_k\, \vec{\gamma})^* J = (J\vec{\gamma})^*\, (J H_k^* J) \qquad (\vec{\gamma} \in \mathbb{C}^k), \qquad J H_k^* J \text{ is upper Hessenberg.}$$

(g) Prove that, for each $\lambda \in \mathbb{C}$, there is a k-vector, for ease of notation also denoted by $\vec{\gamma}_k(\lambda)$, with kth coordinate equal to 1 such that

$$H_k \vec{\gamma}_k(\lambda) = \lambda\, \vec{\gamma}_k(\lambda) - \frac{\det(\lambda\, I_k - H_k)}{\beta_k}\, e_1.$$

Give an expression for the coordinates of this vector and describe the right eigenvectors of H_k. Check that the result is consistent with the one in (5.11).