March 26, 2018

Lecture 4 – Basic Iterative Methods I

A The Power Method

Let $\mathbf{A}$ be an $n \times n$ with eigenvalues $\lambda_1, \ldots, \lambda_n$ counted according to multiplicity.
We assume the eigenvalues to be ordered in absolute decreasing order:

$$|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|.$$

Let $\widetilde{\mathbf{u}} = \mathbf{u}_0$ be a non-zero vector. The **Power Method** iterates as

$$\mathbf{u}_k = \frac{\widetilde{\mathbf{u}}}{\|\widetilde{\mathbf{u}}\|}, \quad \widetilde{\mathbf{u}} = \mathbf{A}\mathbf{u}_k, \quad \rho_k \equiv \mathbf{u}_k^* \mathbf{A}\mathbf{u}_k = \mathbf{u}_k^* \widetilde{\mathbf{u}} \qquad (k = 0, 1, 2, \ldots).$$

With $\mathbf{u}_k = \widetilde{\mathbf{u}}/\|\widetilde{\mathbf{u}}\|$, the iterated vectors are scaled. This might be necessary to avoid overflow (if $|\lambda_1| > 1$) or underflow (if $|\lambda_1| < 1$), but it does not affect convergence. In particular, there is no preference for a specific norm. A scaling with respect to a reference vector, say $\mathbf{w}$ (as $\mathbf{w} = \mathbf{e}_1$, the first standard basis vector) may also be convenient:

$$\mathbf{u}_k = \frac{\widetilde{\mathbf{u}}}{\mathbf{w}^* \widetilde{\mathbf{u}}}, \quad \widetilde{\mathbf{u}} = \mathbf{A}\mathbf{u}_k, \quad \rho_k \equiv \mathbf{w}^* \mathbf{A}\mathbf{u}_k = \mathbf{w}^* \widetilde{\mathbf{u}} \qquad (k = 0, 1, 2, \ldots).$$

Multiples of eigenvectors are also eigenvectors with the same eigenvalues. Therefore, for convergence of eigenvectors it is more appropriate to consider **directional convergence**, i.e., convergence towards 0 of the angle between the approximate eigenvectors and the eigenvector.

Convention 4.1 *We will talk about converging sequences of approximate eigenvectors without explicitly stating that the convergence is supposed to be directionally.*

The following theorem discusses the convergences of the Power Method.

The left eigenvector $\mathbf{y}_1$ associated to the eigenvalue λ_1, i.e., $\mathbf{y}_1^* \mathbf{A} = \lambda_1 \mathbf{y}_1^*$, plays a role for the following reason. Suppose for ease of explanation that there is a basis $\mathbf{x}_1, \ldots, \mathbf{x}_n$ of (right) eigenvectors, $\mathbf{A}\mathbf{x}_j = \lambda_j \mathbf{x}_j$. Then $\mathbf{u}_0$ can be decomposed into eigenvector components: $\mathbf{u}_0 = \alpha_1 \mathbf{x}_1 + \ldots + \alpha_n \mathbf{x}_n$ for appropriate scalars α_j; $\alpha_j \mathbf{x}_j$ is the jth **eigenvector component** of $\mathbf{u}_0$, that is, it is the component of $\mathbf{u}_0$ in the direction of the eigenvector $\mathbf{x}_j$. To find the dominant eigenvector $\mathbf{x}_1$ with the power method, the component of $\mathbf{u}_0$ in the direction of $\mathbf{x}_1$ has to be non-trivial. Since $\mathbf{y}_1 \perp \mathbf{x}_j$ for all $j > 1$ if $\lambda_1 \neq \lambda_j$ $(j > 1)$ (see Exercise 0.22(b)), we have that $\mathbf{y}_1^* \mathbf{u}_0 = \alpha_1 \mathbf{y}_1^* \mathbf{x}_1$. Therefore, the component of interest is non-zero if and only if $\mathbf{y}_1^* \mathbf{u}_0 \neq 0$ (why is $\mathbf{y}_1^* \mathbf{x}_1 \neq 0$?): the $\mathbf{x}_1$-component of $\mathbf{u}_0$ can be computed without computing the other eigenvector components provided the left eigenvector is available. Note that left and right eigenvectors need not to coincide if $\mathbf{A}$ is not Hermitian (actually, if $\mathbf{A}$ is non-normal).

Theorem 4.2 *Assume $|\lambda_1| > |\lambda_2|$. In particular λ_1 is simple.*
Let $\mathbf{x}_1$ be a right eigenvector and $\mathbf{y}_1$ a left eigenvector associated to λ_1.
To simplify notation, put $\mathbf{x} \equiv \mathbf{x}_1$ and $\mathbf{y} \equiv \mathbf{y}_1$. For each $\nu > |\lambda_2|/|\lambda_1|$, we have

$$\mathbf{y}^* \mathbf{u}_0 \neq 0 \quad \Rightarrow \quad \angle(\mathbf{u}_k, \mathbf{x}) \leq \nu^k, \quad |\rho_k - \lambda_1| \leq \nu^k \quad \text{for all } k \text{ large enough.} \tag{4.1}$$

If $\mathbf{A}$ is Hermitian and $\mathbf{x}^* \mathbf{u}_0 \neq 0$, then, for all $k \in \mathbb{N}$,

$$\tan \angle(\mathbf{u}_k, \mathbf{x}) \leq \frac{|\lambda_2|}{|\lambda_1|} \tan \angle(\mathbf{u}_{k-1}, \mathbf{x}) \quad \text{and} \quad |\rho_k - \lambda_1| \leq 2 \|\mathbf{A}\|_2 \sin^2 \angle(\mathbf{u}_k, \mathbf{x}). \tag{4.2}$$

Exercise 4.1. ***Proof of Theorem*** 4.2**.** Let $\mathbf{x}$ and $\mathbf{y}$ be scaled such that $\mathbf{y}^*\mathbf{x} = 1$.

(a) Prove that

$$\frac{\mathbf{u}_0}{\mathbf{y}^*\mathbf{u}_0} = \mathbf{x} + \mathbf{w} \quad \text{for some} \quad \mathbf{w} \perp \mathbf{y}.$$

(b) Prove that for all k

$$\frac{\mathbf{u}_k}{\mathbf{y}^*\mathbf{u}_k} = \mathbf{x} + \mathbf{f}_k, \quad \text{where} \quad \mathbf{f}_k \equiv \frac{1}{\lambda_1^k}\mathbf{A}^k\mathbf{w}.$$

(c) Prove that $\mathbf{A}^k\mathbf{w} \perp \mathbf{y}$.

(d) $\frac{1}{\lambda_1}\mathbf{A}$ maps $\mathbf{y}^\perp$ to $\mathbf{y}^\perp$. Show that the spectral radius of this map is $\frac{|\lambda_2|}{|\lambda_1|}$.

(e) Prove (4.1).

(f) With proper scaling, the error in the approximate eigenvector $\mathbf{u}_k$ can be viewed as iterates in a Power Method: $\mathbf{f}_{k+1} = \frac{1}{\lambda_1}\mathbf{A}\mathbf{f}_k$. In particular, for k large, we have that $\|\mathbf{f}_{k+1}\|_2 \leq \nu\|\mathbf{f}_k\|_2$, and, if $|\lambda_2| > |\lambda_3|$, then $\|\mathbf{f}_{k+1}\|_2 \lesssim \frac{|\lambda_2|}{|\lambda_1|}\|\mathbf{f}_k\|_2$.

(g) Assume $\mathbf{A}$ is Hermitian and $\|\mathbf{x}\|_2 = 1$. Prove that

$$\tan\angle(\mathbf{u}_k, \mathbf{x}) = \|\mathbf{f}_k\|_2 = \|\frac{1}{\lambda_1^k}\mathbf{A}^k\mathbf{w}\|_2 \leq \frac{|\lambda_2|}{|\lambda_1|}\,\|\frac{1}{\lambda_1^{k-1}}\mathbf{A}^{k-1}\mathbf{w}\|_2.$$

With $t \equiv \tan\angle(\mathbf{u}_k, \mathbf{x})$, note that $\sin^2\angle(\mathbf{u}_k, \mathbf{x}) = \frac{t^2}{1+t^2}$. Prove that

$$|\rho_k - \lambda_1| = \frac{|\mathbf{f}_k^*(\mathbf{A} - \lambda_1\mathbf{I})\mathbf{f}_k|}{1 + \|\mathbf{f}_k\|_2^2} \leq \frac{t^2}{1+t^2}\|\mathbf{A} - \lambda_1\mathbf{I}\|_2.$$

(h) Prove (4.2).

Proposition 4.3 *All limit points*[1] *of the sequence* $(\mathbf{u}_k)$ *of normalized vectors generated by the power method are in the span of all eigenvectors with eigenvalue that are equal to* $|\lambda_1|$ *in absolute value.*

Exercise 4.2. **The Power Method.**

(a) Discuss convergence of the Power Method for the following matrices

$$\begin{bmatrix} \lambda & 1 & 0 \\ 0 & \lambda & 1 \\ 0 & 0 & \lambda \end{bmatrix}, \quad \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}.$$

What are the limit points?

(b) Suppose a matrix has real entries and the Power Method is started with a real vector. Can complex eigenvalues be detected?

(c) Let $\mathbf{A}$ be a Jordan block. Show that the sequence $(\mathbf{u}_k)$ of normalized vectors generated by the power method converges (directionally) towards $\mathbf{e}_1$. (Hint. Use a generalisation of (0.9) with $p(\zeta) = \zeta^k$ ($\zeta \in \mathbb{C}$) and show that $\frac{i!}{j!}\frac{p^{(j)}(\lambda)}{p^{(i)}(\lambda)} \to 0$ for $k \to \infty$ and $j < i$.)

(d) Prove Proposition 4.3.

If one eigenpair, $(\lambda_1, \mathbf{x}_1)$, say, has been detected, then deflation techniques, as discussed in the next exercise, can be used to allow detection of other eigenpairs. In exact arithmetic, it suffices to deflate the detected eigenvector $\mathbf{x}_1$ only once (cf., Exercise 4.3.a). However, after

[1] $\mathbf{w}$ is limit point of $(\mathbf{u}_n)$, if there is a sequence (k_j) of positive integers for which $k_j \to \infty$ if $j \to \infty$ and $\|\mathbf{w} - \mathbf{u}_{k_j}\| \to 0$ of $j \to \infty$.

deflation, rounding errors will (initially) introduce a (tiny) component of $\mathbf{x}_1$ in the subsequent process and that will probably lead to a repeated detection of $(\lambda_1, \mathbf{x}_1)$: it may prevent the power method from detecting other eigenpairs. Repeated deflation (as in (b) and (c) of Exercise 4.3) can avoid amplification by the power method of tiny $\mathbf{x}_1$-components.

In lectures to come, we will discuss methods that implicitly incorporate some form of deflation. The insights that we obtain in the next exercise can help to explain why not all methods are equally successful in 'dealing' with rounding errors.

Exercise 4.3. **Deflation.** The Power Method can be used to approximate dominant eigenvalues. In this exercise three methods are given to approximate the eigenvalue λ_2 if λ_1 and associated eigenvector $\mathbf{x}_1$ are known (available). We assume that $|\lambda_1| > |\lambda_2| > |\lambda_3|$ (though not needed, you also may assume that there is a basis of eigenvectors). To simplify notation, we put $\lambda \equiv \lambda_1$ and $\mathbf{x} \equiv \mathbf{x}_1$. Note that $\mathbf{A}$ is a general matrix (in particular, we do not assume $\mathbf{A}$ to be normal or Hermitian, or, more specific, we do not assume that the eigenvectors are mutually orthogonal. The result in Exercise 0.22(b) might be useful, decomposing $\mathbf{A}$ into a Schur form, as in (0.7) and Theorem 0.6, may provide some insight).

(a) Take $\mathbf{u}_0 = (\mathbf{A} - \lambda_1 \mathbf{I})\widetilde{\mathbf{u}}$, where $\widetilde{\mathbf{u}}$ is an arbitrary vector (with a component in the direction of $\mathbf{x}_2$: $\mathbf{y}_2^* \widetilde{\mathbf{u}} \neq 0$. Here $\mathbf{x}_j$ and $\mathbf{y}_j$ are right eigenvector and left eigenvecors, respectively, of $\mathbf{A}$ associated to the eigenvalue λ_j). Show that the Power Method applied to this starting vector leads to an approximation of λ_2 (**Annihilation Technique**).

(b) Show that if the Power Method is applied with the matrix $\mathbf{B}$,

$$\mathbf{B} \equiv \mathbf{A}\left(\mathbf{I} - \frac{\mathbf{x}\,\mathbf{x}^*}{\mathbf{x}^*\mathbf{x}}\right) = \mathbf{A} - \frac{\lambda}{\mathbf{x}^*\mathbf{x}}\mathbf{x}\mathbf{x}^*,$$

one gets an approximation of λ_2 (**Hotelling Deflation**).

The deflation technique that is incorporated in codes for the QR-algorithm (to be discussed in Exercise 4.13 below) can be viewed as an implementation of Hotelling deflation.

(c) Suppose a left eigenvector $\mathbf{y} = \mathbf{y}_1$ with eigenvalue λ_1 is also available: $\mathbf{y}^*\mathbf{A} = \lambda \mathbf{y}^*$. Show that if the Power Method is applied with the matrix $\mathbf{A}'$

$$\mathbf{A}' \equiv \left(\mathbf{I} - \frac{\mathbf{x}\,\mathbf{y}^*}{\mathbf{y}^*\mathbf{x}}\right)\mathbf{A} = \mathbf{A}\left(\mathbf{I} - \frac{\mathbf{x}\,\mathbf{y}^*}{\mathbf{y}^*\mathbf{x}}\right) = \mathbf{A} - \frac{\lambda}{\mathbf{y}^*\mathbf{x}}\mathbf{x}\mathbf{y}^*,$$

then an approximation of λ_2 is obtained. Note that (a) can be proved as an application of (c).

(d) Discuss the advantages and disadvantages of the three methods for computing λ_2 (What is the amount of work per iteration? Keep in mind that $\mathbf{A}$ will be sparse in the applications we are interested in. What can you tell about the stability?)

B Classical iterative methods for linear systems

Gauss–Seidel (GS) and **Gauss–Jacobi** (GJ) are **classical iterative method**s. Their algorithmic representation is given in Alg. 4.1. Both methods cycle through the rows of equations of the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ and compute the correction u_j of the jth coordinate of the approximate solution $\mathbf{x}$ that is needed to satisfy the jth equation. Unfortunately, by meeting the jth equation, the other equations may not be satisfied anymore and repeatedly recycling may be required. In Gauss–Seidel the correction u_j is applied as soon as it is available. In Gauss–Jacobi the correction is postponed until all rows of the equations have been visited.

Both methods have been invented by Gauss (around 1823), but have been reinvented half a century later by Seidel (1874) and Jacobi (1845), respectively. Richardson's method is from 1920. Jacobi applied his method (GJ) to systems that are strongly diagonal dominant (cf., Theorem 4.7), that is, to situations where fast convergence is guaranteed. He obtained such systems by first rotating large off-diagonal elements to zero (Jacobi's method; cf., Exercise 4.14).

In Alg. 4.1, we suppressed the iteration index k. Note that this is in line with the fact that quantities can be replaced (in computer memory, indicated by $\leftarrow$) by their updated version.

GAUSS–SEIDEL

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

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

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

  for $j = 1, \ldots, n$ do

    $u_j = \frac{1}{a_{jj}} r_j$

    $\mathbf{r} \leftarrow \mathbf{r} - \mathbf{A}(u_j \mathbf{e}_j)$

    $x_j \leftarrow x_j + u_j$

  end for

end while

GAUSS–JACOBI

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

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

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

  for $j = 1, \ldots, n$ do

    $u_j = \frac{1}{a_{jj}} r_j$

  end for

  $\mathbf{r} \leftarrow \mathbf{r} - \mathbf{A}\mathbf{u}$

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

end while

RICHARDSON

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

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

Select $\alpha \in \mathbb{C}$

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

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

  $\mathbf{r} \leftarrow \mathbf{r} - \mathbf{A}\mathbf{u}$

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

end while

ALGORITHM 4.1. Gauss–Seidel (at the left), Gauss–Jacobi (in the middle) and Richardson iteration (at the right) for numerically solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$ with residual accuracy *tol*: upon termination, the residual $\mathbf{r}$ of the computed solution $\mathbf{x}$ has norm less than *tol*. Here, $\mathbf{A} = (a_{ij})$ is an $n \times n$ matrix with entries a_{ij}, $\mathbf{r}$ is the residual vector with approximate solution $\mathbf{x}$, and $\mathbf{u}$ is an update vector, with jth coordinate r_j, x_j, and u_j, respectively.

We will say that a method converges for a problem $\mathbf{A}\mathbf{x} = \mathbf{b}$, if the sequence $(\mathbf{x}_n)$ of approximate solutions converges to the solution $\mathbf{x}$ for all initial approximations $\mathbf{x}_0$.

Exercise 4.4. Let $\mathbf{A}$ be a non-singular $n \times n$ matrix. For an $n \times n$ matrix $\mathbf{M}$, put $\mathbf{R} \equiv \mathbf{M} - \mathbf{A}$. With $\mathbf{x}_0 \in \mathbb{C}^n$ and $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$, consider the following two recursions

$$\begin{cases} \text{Solve } \mathbf{M}\mathbf{u}_k = \mathbf{r}_k \text{ for } \mathbf{u}_k, \quad \mathbf{c}_k = \mathbf{A}\mathbf{u}_k, \\ \mathbf{r}_{k+1} = \mathbf{r}_k - \mathbf{c}_k, \quad \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{u}_k \end{cases} \qquad (k = 0, 1, \ldots), \tag{4.3}$$

and

$$\text{Solve } \mathbf{M}\mathbf{x}_{k+1} = \mathbf{R}\mathbf{x}_k + \mathbf{b} \text{ for } \mathbf{x}_{k+1} \qquad (k = 0, 1, \ldots). \tag{4.4}$$

Prove the following claims.

(a) The two **basic iterative method**s (4.3) and (4.4) are equivalent.

(b) The errors and residuals are iterated by

$$\mathbf{x} - \mathbf{x}_{k+1} = \mathbf{M}^{-1}\mathbf{R}(\mathbf{x} - \mathbf{x}_k) \qquad \text{and} \qquad \mathbf{r}_{k+1} = \mathbf{R}\mathbf{M}^{-1}\mathbf{r}_k. \tag{4.5}$$

In view of (4.4), $\mathbf{M}^{-1}\mathbf{R}$ is called the **iteration matrix**. In view of (4.5), this matrix is also called the **error reduction matrix**, while $\mathbf{R}\mathbf{M}^{-1}$ is called the **residual reduction matrix**. Note that these reduction matrices $\mathbf{M}^{-1}\mathbf{R}$ and $\mathbf{R}\mathbf{M}^{-1}$ share the same eigenvalues.

(c) The method converges $\quad \Leftrightarrow \quad \rho(\mathbf{M}^{-1}\mathbf{R}) = \rho(\mathbf{I} - \mathbf{M}^{-1}\mathbf{A}) < 1 \quad \Leftrightarrow \quad \rho(\mathbf{R}\mathbf{M}^{-1}) < 1.$

Write $\mathbf{A}$ as $\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}$, where $\mathbf{D}$ is diagonal, $\mathbf{L}$ is strictly lower triangular, and $\mathbf{U}$ is strictly upper triangular.

(d) Recursion (4.3) (and (4.4)) represents Gauss–Jacobi if $\mathbf{M} = \mathbf{D}$, Gauss–Seidel if $\mathbf{M} = \mathbf{D} - \mathbf{L}$, and Richardson iteration if $\mathbf{M} = \frac{1}{\alpha}\mathbf{I}$

(e) Are there computational advantages involved in the representation here (i.e., in (4.3)) versus those in ALG. 4.1?

Exercise 4.5. **Geometric interpretation of Gauss–Seidel.** Let $\mathbf{A}$ be a non-singular $n \times n$ matrix with ith row $\mathbf{a}_i^* \equiv \mathbf{e}_i^*\mathbf{A}$. Let $\mathbf{b}$ be an n-vector with coordinates β_i.

Consider the hyperplanes $\mathcal{L}_j \equiv \{\mathbf{x} \in \mathbb{C}^n \,\big|\, \mathbf{a}_j^*\mathbf{x} = \beta_j\}$.

(a) $\mathbf{x}$ is at the intersection of all hyperplanes.

(b) Select an $\mathbf{x}_0$. The approximate solution $\mathbf{x}_0^{(1)}$ after the first GS update is at the intersection of the line $\{\mathbf{x}_0 + \alpha\mathbf{e}_1 \,\big|\, \alpha \in \mathbb{C}\}$ and $\mathcal{L}_1$, the next approximate solution, $\mathbf{x}_0^2$, is at the intersection of the line $\{\mathbf{x}_1 + \alpha\mathbf{e}_2 \,\big|\, \alpha \in \mathbb{C}\}$ and $\mathcal{L}_2$, etc.. After n updates $\mathbf{x}_1 \equiv \mathbf{x}_0^{(n)}$.
(c) For $n = 2$, we also consider the system where we switched the first equation $\mathbf{a}_1^*\mathbf{x} = \beta_1$ with the second one $\mathbf{a}_2^*\mathbf{x} = \beta_2$:

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \mathbf{x} = \mathbf{b} = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} \quad \text{and} \quad \widetilde{\mathbf{A}}\mathbf{x} \equiv \begin{bmatrix} a_{21} & a_{22} \\ a_{11} & a_{12} \end{bmatrix} \mathbf{x} = \widetilde{\mathbf{b}} \equiv \begin{bmatrix} \beta_2 \\ \beta_1 \end{bmatrix}.$$

Use the geometrical interpretation of the GS process to show that GS converges for $\mathbf{A}\mathbf{x} = \mathbf{b}$ if and only if GS diverges for $\widetilde{\mathbf{A}}\mathbf{x} = \widetilde{\mathbf{b}}$.

As we learnt in Exercise 4.5, GS does not always converge. However, for some classes of problems that are important in practice, GS does converge. For instance, we have convergence if $\mathbf{A}$ is positive definite. Note that least square problems lead to matrices of this type.[2] Recall that any positive definite matrix is Hermitian (cf., Exercise 0.29).

Theorem 4.4 *Gauss–Seidel converges if* $\mathbf{A}$ *is positive definite.*

Exercise 4.6. ***Proof of Theorem*** 4.4**.** Let $\mathbf{A} = (a_{ij})$ be positive definite. Prove the following claims.

(a) One can write $\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{L}^*$ with $\mathbf{D}$ diagonal and $\mathbf{L}$ strictly lower triangular.

(b) $a_{ii} > 0$ for each $i = 1, \ldots, n$ and $\mathbf{D} - \mathbf{L}$ is non-singular.

Put $\mathbf{G} \equiv (\mathbf{D} - \mathbf{L})^{-1}\mathbf{L}^*$ and $\mathbf{S} \equiv (\mathbf{D} - \mathbf{L})^{-1}\mathbf{A}$. Note that $\mathbf{G}$ is the error reduction matrix of the Gauss–Seidel process (cf., Exercise 4.4).

(c) $\mathbf{G} = \mathbf{I} - (\mathbf{D} - \mathbf{L})^{-1}\mathbf{A} = \mathbf{I} - \mathbf{S}$.

(d) $\mathbf{A} - \mathbf{G}^*\mathbf{A}\mathbf{G} = \mathbf{S}^*(\mathbf{A}\mathbf{S}^{-1} - \mathbf{A} + (\mathbf{S}^*)^{-1}\mathbf{A})\mathbf{S} = \mathbf{S}^*\mathbf{D}\mathbf{S}$.

(e) Let λ be an eigenvalue of $\mathbf{G}$ with eigenvector $\mathbf{x}$. Then $\mathbf{x}^*\mathbf{S}^*\mathbf{D}\mathbf{S}\mathbf{x} > 0$, whence $|\lambda| < 1$.

(f) Theorem 4.4 is correct.

In convergence statements for Gauss–Seidel (GS) and Gauss–Jacobi (GJ) the sign of the matrix entries often plays a role. The following result is useful then. The result is of more general interest.

Theorem 4.5 *Let* $\mathbf{A} = (a_{ij})$ *be an* $n \times n$ *matrix.*
Then, with $|\mathbf{A}| \equiv (|a_{ij}|)$ *and* $\lambda(\mathbf{A}) \equiv \operatorname{argmax}\{|\lambda| \,\big|\, \lambda \in \Lambda(\mathbf{A})\}$, *we have that*

$$|\lambda(\mathbf{A})| \leq \lambda(|\mathbf{A}|). \tag{4.6}$$

Exercise 4.7. ***Proof of Theorem*** 4.5**.** Let $\mathbf{A}$ be as in Theorem 4.5.

(a) Let $\varepsilon > 0$. Use the theorem of Perron–Frobenius (cf., Th. 0.14) to prove that there is an n-vector $\mathbf{y}$ with all coordinates > 0 such that $\mathbf{y}^{\mathrm{T}}(|\mathbf{A}| + \varepsilon\mathbf{1}\mathbf{1}^*) = \lambda(|\mathbf{A}| + \varepsilon\mathbf{1}\mathbf{1}^*)\,\mathbf{y}^{\mathrm{T}}$.

(b) Let $(\lambda, \mathbf{x})$ be an eigenpair of $\mathbf{A}$: $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$.
Show that $|\lambda|\,|\mathbf{x}| \leq |\mathbf{A}|\,|\mathbf{x}| \leq (|\mathbf{A}| + \varepsilon\mathbf{1}\mathbf{1}^*)\,|\mathbf{x}|$. Conclude that $|\lambda| \leq \lambda(|\mathbf{A}| + \varepsilon\mathbf{1}\mathbf{1}^*)$.

(c) Use Theorem 1.13 to prove Theorem 4.5.

In general GS will converge faster than GJ. Nevertheless, for a large (and important) class of matrices GS converges if and only if GJ converges.

[2] The least square solution of $\mathbf{A}\mathbf{x} = \mathbf{b}$ satisfies $\mathbf{A}^*\mathbf{A}\mathbf{x} = \mathbf{A}^*\mathbf{b}$ and $\mathbf{A}^*\mathbf{A}$ is positive definite.

Theorem 4.6 *Let* $\mathbf{A}$ *be an* $n \times n$ *matrix with diagonal* $\mathbf{D}$ *such that* $\mathbf{D} > 0$ *and either* $\mathbf{A} - \mathbf{D} \geq \mathbf{0}$ *or* $\mathbf{D} - \mathbf{A} \geq \mathbf{0}$ *(all diagonal entries are strict positive and all off diagonal entries have the same sign). Then, GS converges* $\Leftrightarrow$ *GJ converges.*

Exercise 4.8. ***Proof of Theorem*** 4.6**.** Let $\mathbf{A} =$ be a non-singular $n \times n$ matrix. Write $\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}$ with $\mathbf{D}$ diagonal, $\mathbf{L}$ strictly lower triangular, and $\mathbf{U}$ strictly upper triangular.

(a) To investigate convergence of GS and GJ we may assume that $\mathbf{D} = \mathbf{I}$. Why?

Assume that $\mathbf{L} + \mathbf{U} \geq 0$.

Let λ and μ be the absolute largest eigenvalue of $(\mathbf{I} - \mathbf{L})^{-1}\mathbf{U}$ and $\mathbf{U} + \mathbf{L}$ respectively. Let $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ be non-trivial n-vectors such that $(\mathbf{U} + \lambda\mathbf{L})\mathbf{x} = \lambda\mathbf{x}$, $(\mathbf{U} + \mathbf{L})\mathbf{z} = \mu\mathbf{z}$, and $\mathbf{y}^*(\mathbf{U} + \mathbf{L}) = \mu\mathbf{y}^*$.

(b) We may assume that $\mathbf{x} \geq 0$, $\mathbf{y} \geq 0$, and $\mathbf{z} \geq 0$, $\lambda > 0$ and $\mu > 0$. Why?

(c) $\lambda\mathbf{y}^*\mathbf{x} = \mathbf{y}^*(\mathbf{U} + \lambda\mathbf{L})\mathbf{x} = \mu\mathbf{y}^*\mathbf{x} + (\lambda - 1)\mathbf{y}^*\mathbf{L}\mathbf{x}$ and $\mathbf{y}^*(\mathbf{U} + \mathbf{L})\mathbf{x} = \mathbf{y}^*\mathbf{U}\mathbf{x} + \frac{\lambda - \mu}{\lambda - 1}\mathbf{y}^*\mathbf{x} = \mu\mathbf{y}^*\mathbf{x}$. Whence,

$$\lambda\frac{\mu - 1}{\lambda - 1} = \frac{\mathbf{y}^*\mathbf{U}\mathbf{x}}{\mathbf{y}^*\mathbf{x}} \geq 0.$$

(d) Conclude that $\lambda < 1 \Rightarrow \mu \leq 1$ and $\lambda > 1 \Rightarrow \mu \geq 1$.

(e) Assume that $\lambda < 1$ and $\mu = 1$. For $\rho > 1$, consider $\rho\mathbf{U}$ and $\rho\mathbf{L}$ instead of $\mathbf{U}$ and $\mathbf{L}$, respectively. Then, the associated μ is > 1, whereas the associated λ is < 1 for $\rho \approx 1$ (use the fact that the absolute largest eigenvalue depends continuously on ρ). Conclude that $\lambda < 1 \Rightarrow \mu < 1$.

(f) Prove Theorem 4.6.

An $n \times n$ matrix $\mathbf{A} = (a_{ij})$ is strict **diagonal dominant** if $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$ for all i.

Theorem 4.7 *Both GS and GJ converge for matrices that are strict diagonal dominant.*

Exercise 4.9. ***Proof of Theorem*** 4.7**.** For an $n \times n$ strict diagonal dominant matrix $\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}$, with $\mathbf{D}$ diagonal, $\mathbf{L}$ strict lower triangular and $\mathbf{U}$ is strict upper tridiagonal, prove the following claims.

(a) $\mathbf{A}$ is non-singular.

(b) Put $\widetilde{\mathbf{L}} \equiv \mathbf{D}^{-1}\mathbf{L}$ and $\widetilde{\mathbf{U}} \equiv \mathbf{D}^{-1}\mathbf{U}$. Then $\| |\widetilde{\mathbf{L}}| + |\widetilde{\mathbf{U}}| \|_\infty < 1$ and $|\lambda(\widetilde{\mathbf{L}} + \widetilde{\mathbf{U}})| < 1$.

(c) GJ converges.

(d) $|\lambda((\mathbf{I} - \widetilde{\mathbf{L}})^{-1}\widetilde{\mathbf{U}})| \leq \lambda((\mathbf{I} - |\widetilde{\mathbf{L}}|)^{-1}|\widetilde{\mathbf{U}}|) < 1$. (Hint: use the results of Exercise 4.8.)

(e) GS converges.

Successive over-relaxation (**SOR**) is the modification of Gauss–Seidel, where a '**relaxation parameter**' ω is selected before the start of the iteration process and each u_j is replaced by ωu_j. With $\omega > 1$, we talk about over-relaxation, with $\omega < 1$ we have under-relaxation. Often SOR converges faster than GS for some appropriate parameter $\omega \in (1, 2)$. What the best value of ω is, is problem dependent.

Gauss–Seidel and SOR, cycle repeatedly through all rows of the system from top to bottom (the 'for $j = 1, \ldots, n$ do' loop). Of course, we can also cycle from bottom to top or reverse the order alternatingly. The SOR version, where the order of running through the rows of the system is reversed after each cycle is called **Symmetric SOR** (**SSOR**). Note that the word 'symmetric' refers to a symmetric way of applying the process and not to a property of the matrix.

Exercise 4.10. Let $\mathbf{A}$ be a non-singular $n \times n$ matrix. Write $\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}$ with $\mathbf{D}$ diagonal, $\mathbf{L}$ strictly lower triangular, and $\mathbf{U}$ strictly upper triangular.

(a) Show that SOR is a basic iterative method with $\mathbf{M} = \frac{1}{\omega}\mathbf{D} - \mathbf{L}$.

(b) Adapt the arguments in Exercise 4.6 to show that, for a positive definite matrix $\mathbf{A}$, we have that SOR converges $\Leftrightarrow$ $\omega \in (0, 2)$.

(c) Show that SSOR is a basis iterative method with

$$\mathbf{M} = (\tfrac{1}{\omega}\mathbf{D} - \mathbf{L})([\tfrac{2}{\omega} - 1]\mathbf{D})^{-1}(\tfrac{1}{\omega}\mathbf{D} - \mathbf{U}).$$

Exercise 4.11. Consider the basic iterative method ($\mathbf{A} = \mathbf{M} - \mathbf{R}$)

$$\mathbf{M}\mathbf{x}_{k+1} = \mathbf{R}\mathbf{x}_k + \mathbf{b}.$$

(a) Explain how the residual can be computed from the vectors $\mathbf{R}\mathbf{x}_k + \mathbf{b}$ with simple vector updates (no additional matrix vector multiplications).

However, we are interested in having small errors and the residuals may give a wrong impression of the error. We would therefore like to use a (cheap) termination criterion, based on the true error instead of on the residual. Below we derive such a criterion.

(b) Show that the spectral radius of the **iteration matrix** $\mathbf{G} \equiv \mathbf{M}^{-1}\mathbf{R}$ approximately satisfies

$$\rho(\mathbf{G}) \approx \frac{\|\mathbf{x}_{k+1} - \mathbf{x}_k\|}{\|\mathbf{x}_k - \mathbf{x}_{k-1}\|}.$$

(c) Show that if $\rho(\mathbf{G})$ is known, an estimate for the error is given by

$$\|\mathbf{x} - \mathbf{x}_k\|_2 \leq \frac{\rho(\mathbf{G})}{1 - \rho(\mathbf{G})}\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_2.$$

(Hint: bound $\|\mathbf{x}_j - \mathbf{x}_k\|_2$ in terms of $\rho(\mathbf{G})$ and $\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_2$. Then take the limit $\mathbf{x} = \lim_{j\to\infty} \mathbf{x}_j$.)

C Solvers of eigenvalue problems for non-high dimensional matrices

Let $\mathbf{A}$ be an $n \times n$ matrix.
Since, for scalars σ, the **shifted matrix** $\mathbf{A} - \sigma\mathbf{I}$ and **shift-and-inverted matrix** $(\mathbf{A} - \sigma\mathbf{I})^{-1}$ have the same eigenvectors as $\mathbf{A}$, they can also be used to compute eigenvectors of $\mathbf{A}$. σ is called a **shift** (in $(\mathbf{A} - \sigma\mathbf{I})^{-1}$ it is also sometimes called a **pole**). For well choosen shifts, the power method with such a modified matrix may converge faster or may detect other eigenvectors (eigenvectors that are dominant with respect to the modified matrix): for instance, the eigenvector of $\mathbf{A}$ with eigenvalue closest to σ will be dominant for the shift-and-inverted matrix. The power method with the shift-and-inverted matrix is called **Shift-and-Invert iteration** or also Shift-and-Invert power method and **Wielandt iteration**.

Assume some approximate eigenvector $\mathbf{u}$ is available. Consider the **Rayleigh quotient**

$$\rho(\mathbf{u}) \equiv \frac{\mathbf{u}^*\mathbf{A}\mathbf{u}}{\mathbf{u}^*\mathbf{u}}.$$

In some sense the Rayleigh quotient forms the best approximate eigenvalue associated with the approximate eigenvector. It gives the smallest residuals (why?):

$$\rho(\mathbf{u}) = \text{argmin}_\vartheta \|\mathbf{A}\mathbf{u} - \vartheta\,\mathbf{u}\|_2.$$

The following 'variant' of Shift-and-Invert is called **Rayleigh Quotient Iteration** (RQI). Select a $\widetilde{\mathbf{u}} = \mathbf{u}_0 \in \mathbb{C}^n,\ \mathbf{u} \neq \mathbf{0}$. For $k = 0, 1, \ldots$ do

$$\mathbf{u}_k = \frac{\widetilde{\mathbf{u}}}{\|\widetilde{\mathbf{u}}\|_2}, \quad \text{solve } (\mathbf{A} - \lambda^{(k)}\mathbf{I})\widetilde{\mathbf{u}} = \mathbf{u}_k \text{ for } \widetilde{\mathbf{u}}, \quad \lambda^{(k)} = \rho(\mathbf{u}_k).$$

In Shift-and-Invert, the shift is fixed throughout the iteration. Selecting the shift close to an eigenvalue guaranties fast convergence towards that eigenvalue (why?). Rayleigh Quotient

THE QR-ALGORITHM

Select $\mathbf{U}$ unitary. $\mathbf{S} = \mathbf{U}^*\mathbf{A}\mathbf{U}$,
$m = \mathtt{size}(\mathbf{A}, 1)$, $N = [1 : m]$, $\mathbf{I} = \mathbf{I}_m$.
repeat until $m = 1$
  1) Select the Wilkinson shift σ
  2) $[\mathbf{Q}, \mathbf{R}] = \mathtt{qr}(\mathbf{S}(N, N) - \sigma\,\mathbf{I})$
  3) $\mathbf{S}(N, N) = \mathbf{R}\mathbf{Q} + \sigma\mathbf{I}$
  4) $\mathbf{U}(:, N) \leftarrow \mathbf{U}(:, N)\mathbf{Q}$
  5) if $|\mathbf{S}(m, m-1)| \le \varepsilon\,|\mathbf{S}(m, m)|$ %% *Deflate*
       $\mathbf{S}(m, m-1) = 0$
       $m \leftarrow m-1$, $N \leftarrow [1 : m]$, $\mathbf{I} \leftarrow \mathbf{I}_m$
     end if
end repeat

ALGORITHM 4.2. The QR-algorithm for computing the Schur decomposition $\mathbf{AU} = \mathbf{US}$ of a general square matrix $\mathbf{A}$ with accuracy ε. The initializing $\mathbf{U}$ is selected such that the initial $\mathbf{S}$ is upper Hessenberg. Upon convergence we have that $\mathbf{AU} = \mathbf{US}$ with $\mathbf{U}$ unitary and $\mathbf{S}$ upper triangular. Note that we used MATLAB conventions to indicate submatrices. As the MATLAB function routine `qr`, `qr` on line 2) refers to a sub-algorithm that computes the QR factors $\mathbf{Q}$ and $\mathbf{R}$ of the QR-decomposition. The Wilkinson shift is an eigenvalue of the 2×2 right lower block $\mathbf{S}([m-1, m], [m-1, m])$ of the 'active' part $S(N, N)$ of $\mathbf{S}$.

Iteration "tries" to improve on this approach by taking the shift in each step equal to the 'best' eigenvalue approximation that is available in that step, that is, it takes as shift the Rayleigh quotient of the approximate eigenvector.

Exercise 4.12. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

(a) Discuss the convergence of the Power Method and the **inverse Power Method**, i.e., iterate with $\mathbf{A}^{-1}$.

(b) Discuss the convergence of the **shifted Power Method** and the shift-and-inverted Power Method with shift $\sigma = 1.5$, i.e., iterate with $\mathbf{A} - \sigma\mathbf{I}$ and with $(\mathbf{A} - \sigma\,\mathbf{I})^{-1}$.

(c) Discuss the convergence of Rayleigh Quotient Iteration for $\mathbf{u}_0 = \mathbf{e}_1$ and for $\mathbf{u}_0 = \mathbf{e}_1 + \mathbf{e}_2$.

(d) Can Rayleigh Quotient Iteration be used to compute complex (non-real) eigenvalues if $\mathbf{A}$ is real and $\mathbf{u}_0$ is real?

The QR-algorithm. Let $\mathcal{A}$ be a linear map from $\mathbb{C}^n$ to $\mathbb{C}^n$.
We denote the matrix of this map $\mathcal{A}$ represented with respect to the standard basis by $\mathbf{A}$.
Let $\mathbf{A}_0$ be the matrix of $\mathcal{A}$ with respect to an orthonormal basis $\mathbf{u}_1, \ldots, \mathbf{u}_n$ (in this exercise, we use the same basis for image space as for domain space). Then, $\mathbf{AU}_0 = \mathbf{U}_0\mathbf{A}_0$, where $\mathbf{U}_0$ is the unitary matrix $\mathbf{U}_0 \equiv [\mathbf{u}_1, \ldots, \mathbf{u}_n]$ with columns the basic vectors $\mathbf{u}_j$. Note that a switch of basis does not change the eigenvalues (nor the eigenvectors. It only changes the way these vectors are beings represented).

The essential steps of the **QR-algorithm** that we will discuss in the following exercise can be described as:

$$\begin{aligned} &\text{for } k = 0, 1, \ldots \text{ do} \\ &\quad \bullet \text{ select a shift } \sigma_k, \\ &\quad \bullet \text{ factorise } \quad \mathbf{A}_k - \sigma_k \mathbf{I} = \mathbf{Q}_k \mathbf{R}_k, \\ &\quad \bullet \text{ multiply } \quad \mathbf{A}_{k+1} \equiv \mathbf{R}_k \mathbf{Q}_k + \sigma_k \mathbf{I}, \\ &\quad \bullet \text{ multiply } \quad \mathbf{U}_{k+1} \equiv \mathbf{U}_k \mathbf{Q}_k. \end{aligned} \tag{4.7}$$

The factorisation in the second sub-step is a QR-factorisation, that is, $\mathbf{Q}_k$ is unitary and $\mathbf{R}_k$ is upper triangular. An extended version of the QR-algorithm (in pseudo code), including deflation, can be found in Alg. 4.2. Note that the QR-algorithm exploits a QR-decomposition in each step (to be more specific, in the second sub-step; do not confuse the naming of 'QR-algorithm' and 'QR-decomposition'!).

In the next exercise we will learn that, with a proper shift selection strategy, we will have convergence towards a Schur decomposition of $\mathbf{A}$ (cf., (0.7) and Exercise 0.17): the $\mathbf{U}_k$ will converge to a unitary matrix $\mathbf{U}$ and the $\mathbf{A}_k$ to an upper triangular matrix $\mathbf{S}$ such that

$$\mathbf{AU} = \mathbf{US}.$$

Note that the first three sub-steps do not rely on the fourth. Therefore, there is no need to perform this sub-step if we are interested in eigenvalues only: they will eventually show up at the diagonal of $\mathbf{A}_k$.

Exercise 4.13. **The QR-algorithm.** Consider the above situation.

(a) Show that $\mathbf{AU}_0\mathbf{Q}_0 = \mathbf{U}_0\mathbf{Q}_0\mathbf{A}_1$. Interpret $\mathbf{A}_1$ as the matrix of $\mathcal{A}$ with respect to the basis $\mathbf{u}_1^{(1)}, \ldots, \mathbf{u}_n^{(1)}$ of columns of $\mathbf{U}_1 \equiv \mathbf{U}_0\mathbf{Q}_0$.

(b) Show that $(\mathbf{A}_0 - \sigma_0\mathbf{I})^*\mathbf{Q}_0 = \mathbf{R}_0^*$.

(c) Show that the first column of $\mathbf{U}_1$ can be viewed as arising from an application of one step of the shifted Power Method applied to the first column of $\mathbf{U}_0$, while for the last column of $\mathbf{U}_1$ one step of Shift-and-Invert (by what matrix?) has been applied to the last column of $\mathbf{U}_0$. Note that this has been achieved without doing the inversion explicitly!

(d) Interpret the left top element $a_{11}^{(1)}$ of $\mathbf{A}_1$ and the right bottom element $a_{nn}^{(1)}$ as Rayleigh quotients for vectors in the Power Methods of (c).

(e) Use the interpretation in (a), to explain how the QR-algorithm incorporates the shifted Power Method as well as the shift-and-invert Power Method.

(f) Suggest a choice for σ_k to obtain fast (quadratic) convergence. How do you compute σ_k?

To avoid the type of stagnation as discussed in Exercise 4.12(c), a **Wilkinson shift** is selected, that is an eigenvalue of the 2×2 right lower block of $\mathbf{A}_0$ (in step 0 and of $\mathbf{A}_k$ is step k).

Suppose $\mathbf{A}_0 = (a_{ij}^{(0)})$ is upper Hessenberg (that is, $a_{ij}^{(0)} = 0$ if $i > j + 1$).

(g) Prove that $\mathbf{Q}_0$ and $\mathbf{A}_1$ are upper Hessenberg.

(h) Show that the following expressions for the norm of the residuals are correct

$$\|\mathbf{A}_0\mathbf{q}_1^{(0)} - a_{11}^{(1)}\mathbf{q}_1^{(0)}\|_2 = |a_{21}^{(1)}|, \qquad \|\mathbf{A}_0^*\mathbf{q}_n^{(0)} - \overline{a}_{nn}^{(1)}\mathbf{q}_n^{(0)}\|_2 = |a_{n,n-1}^{(1)}|.$$

Assuming Householder reflections (see Exercise 3.16) or Givens rotations (see Exercise 3.5) are used for orthonormalisations (in the QR-decompositions), then the QR-algorithm is *backward stable*, that is, if $\mathbf{U}_k$ is the unitary matrix computed at step $k-1$, $\mathbf{U}_k = \mathbf{U}_0\mathbf{Q}_0\mathbf{Q}_1 \cdots \mathbf{Q}_{k-1}$, and $\mathbf{S}_k$ is the upper Hessenberg matrix, $\mathbf{S}_k = \mathbf{A}_k$, then for some perturbation matrix Δ of size of $\mathcal{O}(\varepsilon)$ (i.e., $\|\Delta\|_2 = \mathcal{O}(\varepsilon)$) we have that

$$(\mathbf{A} + \Delta)\mathbf{U}_k = \mathbf{U}_k\mathbf{S}_k :$$

JACOBI'S METHOD

$\mathbf{A}^{(0)} \equiv \mathbf{A}$.

`For` $k = 1, 2, \ldots$ `do`

1) $(i,j) = \texttt{argmax}\{|A_{ij}^{(k-1)}| \,\big|\, (i,j),\ i > j\}$.

2) `Construct a Givens rotation` $\mathbf{G}$ `such that` $A_{ij}^{(k)} = 0$, `where` $\mathbf{A}^{(k)} \equiv \mathbf{G}^* \mathbf{A}^{(k-1)} \mathbf{G}$.

ALGORITHM 4.3. Jacobi's method iterates an Hermitian matrix $\mathbf{A}$ to diagonal by recursively rotating large off-diagonal elements to zero.

the computed quatities are the exact ones of a slightly perturbed matrix $\mathbf{A}$.[3] The same holds for the unitary matrix $\mathbf{U}$, $\mathbf{U} = \mathbf{U}_k$, and the upper triangular matrix $\mathbf{S}$, $\mathbf{S}$ is the upper triangular part of $\mathbf{S}_k$, at termination when the lower diagonal of $\mathbf{S}_k$ is of $\mathcal{O}(\varepsilon)$.

As the QR-algorithm, the methods in the exercises below are for computing eigenvalues of low dimensional matrices (of dimension at most a few thousand). However, unlike the QR-algorithm, they are for Hermitian matrices only and they are not based on the Power Method. They are mentioned here for completes and since the methods to be discussed later in this course project high-dimensional problems to low dimensional ones. The low dimensional problems are solved with methods as the QR-algorithm or the ones below.

Exercise 4.14. **Jacobi's method for diagonalising Hermitian matrices.** Let $\mathbf{A}$ be an $n \times n$ Hermitian.

For notational convenience, first take $(i,j) = (2,1)$. From Exercise 3.5(c) we know that there is a Givens rotation that rotates in the $(1,2)$-plane such that the $(2,1)$-entry of $\mathbf{G}^*\mathbf{AG}$ is 0.

(a) Show that $\|\mathbf{A}(J,[1,2])\mathbf{G}([1,2],[1,2])\|_{\mathrm{F}} = \|\mathbf{A}(J,[1,2])\|_{\mathrm{F}}$. Here, $J \equiv [3,4,\ldots,n]$ and we used MATLAB notation to denote sub-matrices. Conclude that the Frobenius norm of the part of the matrices $\mathbf{A}$ and $\mathbf{G}^*\mathbf{AG}$ outside the 2×2 left upper block are the same.

Let $\nu(\mathbf{A})$ be the Frobenius norm of the off-diagonal part of $\mathbf{A}$ ($\nu(\mathbf{A}) \equiv \|\mathbf{A} - \mathrm{diag}(\mathbf{A})\|_{\mathrm{F}}$).

(b) Prove that $\nu(\mathbf{G}^*\mathbf{AG}) = \sqrt{\nu(\mathbf{A})^2 - 2|A_{2,1}|^2}$.

Consider the procedure in ALG. 4.3 (**Jacobi's method**) for constructing a sequence of matrices $\mathbf{A}^{(k)} = (A_{ij}^{(k)})$.

(c) Prove that $\nu(\mathbf{A}^{(k)}) \to 0$ for $k \to \infty$. Show that $(\mathbf{A}^{(k)})$ converges to a diagonal matrix, say $\mathbf{D}$, and $\mathbf{AQ} = \mathbf{QD}$ for some $n \times n$ unitary matrix $\mathbf{Q}$: Jacobi's method converges to an eigenvalue decomposition of $\mathbf{A}$.

(d) Now, assume $\mathbf{A}$ is not Hermitian and let $\nu(\mathbf{A})$ be the Frobenius norm of the strict lower triangular part of $\mathbf{A}$. Can the above procedure be use to compute the Schur form of $\mathbf{A}$?

Jacobi used in 1842 a few steps of this method to make a symmetric matrix a bit more 'diagonal dominant' (that is, to reduce the Frobenius norm of the off-diagonal elements). In the 1950's it was used for computing eigenvalues of symmetric matrices. After introduction of the QR-algorithm, this approached lost importance. The method was revived after the introduction of parallel computers: Jacobi's method can be easily parallelised if step 1) is replaced by a cyclic selection procedure, whereas the QR-algorithm is sequential.

Exercise 4.15. **Sturm sequences.** Let $\mathbf{T}$ be an Hermitian tri-diagonal $n \times n$ matrix with $\alpha_1, \alpha_2, \ldots$ on its diagonal and $\beta_1, \beta_2, \ldots$ on its first upper co-diagonal. Note that $\alpha_j \in \mathbb{R}$

[3]The algorithm can be *forward unstable*: entries of the obtained matrices can have no digit in common with the corresponding matrix entries of the exact results, i.e., the ones that would have been obtained in exact arithmetic.

and $\bar{\beta}_1, \bar{\beta}_2, \ldots$ are on the first lower co-diagonal. Assume $\mathbf{T}$ is unreduced, i.e., $\beta_j \neq 0$ for all $j = 1, \ldots, n-1$.

Let p_k be the characteristic polynomial of the $k \times k$ left upper block T_k of $\mathbf{T}$. Note that the zeros of p_k are the eigenvalues of T_k. In particular, these zeros are real.

(a) Prove that, with $p_0(\zeta) = 1$,

$$p_1(\zeta) = \zeta - \alpha_1, \quad p_{k+1}(\zeta) = (\zeta - \alpha_{k+1})p_k(\zeta) - |\beta_k|^2 p_{k-1}(\zeta) \qquad (\zeta \in \mathbb{C},\ k = 1, 2, \ldots). \quad (4.8)$$

(b) Prove that p_{k+1} and p_k do not share a common zero.
(Hint. If they do, then p_k and p_{k-1} have a common zero).

(c) Suppose that the zeros of p_{k-1} **interlace** the zeros of p_k, that is, in between two zeros of p_k, there is a zero of p_k. Hence, if $\lambda_{1,k} < \lambda_{2,k} < \ldots < \lambda_{k,k}$ are the zeros of p_k ordered in increasing magnitude, then

$$\lambda_{1,k} < \lambda_{1,k-1} < \lambda_{2,k} < \lambda_{2,k-1} < \ldots < \lambda_{k-1,k-1} < \lambda_{k,k}$$

Note that $p_{k+1}(\lambda) > 0$ if $\lambda \to \infty$. Show that p_{k+1} and p_{k-1} have opposite signs at the zeros of p_k. Conclude that the zeros of p_k interlace the zeros of p_{k+1}.
Show that the zeros of consecutive p_ks interlace. In particular, all zeros of a p_k are simple (if $p_k(\lambda) = 0$, then $p_k'(\lambda) \neq 0$).

(d) Let $\alpha \in \mathbb{R}$, α not an eigenvalue of $\mathbf{T}$. The sequence of signs of $p_0(\alpha), p_1(\alpha), p_2(\alpha), \ldots, p_n(\alpha)$ is a **Sturm sequence** at α. Let $\chi(\alpha)$ be the number of changes in signs in the Sturm sequence at α. Prove that there are exactly $\chi(\alpha)$ eigenvalues of $\mathbf{T}$ in (α, ∞). What is a sign change if a $p_j(\alpha) = 0$?

Let $\beta \in (\alpha, \infty)$, $\beta \notin \Lambda(\mathbf{T})$. Prove that the number of eigenvalues of $\mathbf{T}$ in the interval (α, β) equals $\chi(\alpha) - \chi(\beta)$.

Note that the number of sign changes comes for free if (4.8) is used to compute the value of $p_n(\alpha)$. The Sturm sequence approach give a way to compute the 'distribution' of eigenvalues for Hermitian tri-diagonal matrices.

Sturm sequences can be formed for very high dimensional matrices, provided they are Hermitian tri-diagonal. These matrices are formed by the Lanczos method (to be discussed in Lecture 7).