Chapter 7

Vector iteration (power method)

7.1 Simple vector iteration

In this chapter we consider the simplest method to compute a single extremal eigenvalue, called *vector iteration* or *power method* [2, 5]. Let $A \in \mathbb{F}^{n\times n}$. Starting with an arbitrary initial vector $\mathbf{x}^{(0)} \in \mathbb{F}^n$ we form the vector sequence $\left\{\mathbf{x}^{(k)}\right\}_{k=0}^{\infty}$ by defining

$$\mathbf{x}^{(k)} := A\mathbf{x}^{(k-1)}, \qquad k = 1, 2, \ldots \tag{7.1}$$

Clearly,

$$\mathbf{x}^{(k)} := A^k\mathbf{x}^{(0)}. \tag{7.2}$$

The hope is that the $\mathbf{x}^{(k)}$ converge to an eigenvector associated with the eigenvalue of largest magnitude. As we are interested only in the direction but not in the length of the eigenvector, there is no need to normalize the iterates in (7.1), well at least *in theory*. In practice, $\mathbf{x}^{(k)}$ may either underflow (if $\|A\| < 1$) or overflow (if $\|A\| > 1$) for large k. Thus, one usually adds a normalization step to (7.1), leading to Algorithm 7.1.

Algorithm 7.1 Simple vector iteration (power method)

1: Choose a starting vector $\mathbf{x}^{(0)} \in \mathbb{F}^n$ with $\|\mathbf{x}^{(0)}\| = 1$.
2: $k = 0$.
3: **repeat**
4: $\quad k := k + 1$;
5: $\quad \mathbf{y}^{(k)} := A\mathbf{x}^{(k-1)}$;
6: $\quad \mu_k := \|\mathbf{y}^{(k)}\|$;
7: $\quad \mathbf{x}^{(k)} := \mathbf{y}^{(k)}/\mu_k$;
8: **until** a convergence criterion is satisfied

The vectors $\mathbf{x}^{(k)}$ generated by Algorithm 7.1 have all norm (length) one. That is, $\left\{\mathbf{x}^{(k)}\right\}_{k=0}^{\infty}$ is a sequence on the unit sphere in $\mathbb{F}^n$.

Let $A = XJY^*$ be the Jordan normal form of A with $Y^* := X^{-1}$, see Section 2.8. Then,

$$Y^*\mathbf{x}^{(k)} := JY^*\mathbf{x}^{(k-1)} \qquad \text{and} \qquad Y^*\mathbf{x}^{(k)} := J^kY^*\mathbf{x}^{(0)}, \tag{7.3}$$

respectively. If the sequence $\left\{\mathbf{x}^{(k)}\right\}_{k=0}^{\infty}$ converges to $\mathbf{x}_*$ then the sequence $\left\{\mathbf{y}^{(k)}\right\}_{k=0}^{\infty}$ with $\mathbf{y}^{(k)} = Y^*\mathbf{x}^{(k)}$ converges to $\mathbf{y}_* = Y^*\mathbf{x}_*$. By consequence, for the convergence analysis, we may assume without loss of generality (w.l.o.g.) that A is a Jordan block matrix.

7.2 Angles between vectors

Let $\mathbf{q}_1$ and $\mathbf{q}_2$ be unit vectors, cf. Fig. 7.1. The length of the orthogonal projection of $\mathbf{q}_2$

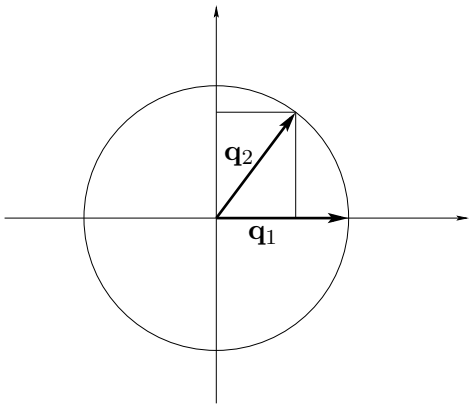

Figure 7.1: Angle between vectors $\mathbf{q}_1$ and $\mathbf{q}_2$

on span$\{\mathbf{q}_1\}$ is given by

$$c := \|\mathbf{q}_1\mathbf{q}_1{}^*\mathbf{q}_2\| = |\mathbf{q}_1{}^*\mathbf{q}_2| \leq 1. \tag{7.4}$$

The length of the orthogonal projection of $\mathbf{q}_2$ on span$\{\mathbf{q}_1\}^\perp$ is

$$s := \|(\mathbf{I} - \mathbf{q}_1\mathbf{q}_1{}^*)\mathbf{q}_2\|. \tag{7.5}$$

As $\mathbf{q}_1\mathbf{q}_1^*$ is an orthogonal projection we have by Pythagoras' formula that

$$1 = \|\mathbf{q}_2\|^2 = \|\mathbf{q}_1\mathbf{q}_1{}^*\mathbf{q}_2\|^2 + \|(\mathbf{I} - \mathbf{q}_1\mathbf{q}_1{}^*)\mathbf{q}_2\|^2 = s^2 + c^2. \tag{7.6}$$

Alternatively, we can conclude from (7.5) that

$$\begin{aligned} s^2 &= \|(\mathbf{I} - \mathbf{q}_1\mathbf{q}_1{}^*)\mathbf{q}_2\|^2 \\ &= \mathbf{q}_2{}^*(\mathbf{I} - \mathbf{q}_1\mathbf{q}_1{}^*)\mathbf{q}_2 \\ &= \mathbf{q}_2{}^*\mathbf{q}_2 - (\mathbf{q}_2{}^*\mathbf{q}_1)(\mathbf{q}_1{}^*\mathbf{q}_2) \\ &= 1 - c^2 \end{aligned} \tag{7.7}$$

So, there is a number, say, ϑ, $0 \leq \vartheta \leq \frac{\pi}{2}$, such that $c = \cos\vartheta$ and $s = \sin\vartheta$. We call this uniquely determined number ϑ the **angle** between the vectors $\mathbf{q}_1$ and $\mathbf{q}_2$:

$$\vartheta = \angle(\mathbf{q}_1, \mathbf{q}_2).$$

The generalization to arbitrary vectors is straightforward.

Definition 7.1 The **angle** θ between two nonzero vectors $\mathbf{x}$ and $\mathbf{y}$ is given by

$$\vartheta = \angle(\mathbf{x}, \mathbf{y}) = \arcsin\left(\left\|\left(I - \frac{\mathbf{x}\mathbf{x}^*}{\|\mathbf{x}\|^2}\right)\frac{\mathbf{y}}{\|\mathbf{y}\|}\right\|\right) = \arccos\left(\frac{|\mathbf{x}^*\mathbf{y}|}{\|\mathbf{x}\|\|\mathbf{y}\|}\right). \tag{7.8}$$

When investigating the convergence behaviour of eigensolvers we usually show that the angle between the approximating and the desired vector tends to zero as the number of iterations increases. In fact it is more convenient to work with the sine of the angle.

In the formulae above we used the projections P and $I-P$ with $P = \mathbf{q}_1\mathbf{q}_1{}^*$. We would have arrived at the same point if we had exchanged the roles of $\mathbf{q}_1$ and $\mathbf{q}_2$. As

$$\|\mathbf{q}_1\mathbf{q}_1^*\mathbf{q}_2\| = \|\mathbf{q}_2\mathbf{q}_2^*\mathbf{q}_1\| = |\mathbf{q}_2^*\mathbf{q}_1|$$

we get

$$\|(I - \mathbf{q}_1\mathbf{q}_1^*)\mathbf{q}_2\| = \|(I - \mathbf{q}_2\mathbf{q}_2^*)\mathbf{q}_1\|.$$

This immediately leads to

Lemma 7.2 $\sin\angle(\mathbf{q}_1, \mathbf{q}_2) = \|\mathbf{q}_1\mathbf{q}_1^* - \mathbf{q}_2\mathbf{q}_2^*\|$.

7.3 Convergence analysis

Let us now assume that A has Jordan block form,

$$A = \begin{bmatrix} \lambda_1 & \mathbf{0}^* \\ \mathbf{0} & J_2 \end{bmatrix} \tag{7.9}$$

with eigenvalues

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|. \tag{7.10}$$

Then, the eigenvector of A corresponding to its largest eigenvalue λ_1 is $\mathbf{e}_1$. We will now show that the iterates $\mathbf{x}^{(k)}$ converge to $\mathbf{e}_1$. More precisely, we will show that the angle $\angle(\mathbf{x}^{(k)}, \mathbf{e}_1)$ between $\mathbf{x}^{(k)}$ and $\mathbf{e}_1$ goes to zero with $k \to \infty$. Let

$$\mathbf{x}^{(k)} = \begin{pmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_n^{(k)} \end{pmatrix} =: \begin{pmatrix} x_1^{(k)} \\ \mathbf{x}_2^{(k)} \end{pmatrix}$$

with $\|\mathbf{x}^{(k)}\| = 1$. Then,

$$\sin\vartheta^{(k)} := \sin(\angle(\mathbf{x}^{(k)}, \mathbf{e}_1)) = \|(I - \mathbf{e}_1\mathbf{e}_1^*)\mathbf{x}^{(k)}\| = \|\mathbf{x}_2^{(k)}\| = \sqrt{\sum_{i=2}^{n} |x_i^{(k)}|^2}.$$

If we omit the normalization $\|\mathbf{x}^{(k)}\| = 1$, which we will do for convenience, then this becomes

$$\sin\vartheta^{(k)} := \sin(\angle(\mathbf{x}^{(k)}, \mathbf{e}_1)) = \frac{\|\mathbf{x}_2^{(k)}\|}{\|\mathbf{x}^{(k)}\|} = \sqrt{\frac{\sum_{i=2}^{n} |x_i^{(k)}|^2}{\sum_{i=1}^{n} |x_i^{(k)}|^2}}.$$

This means that for the convergence analysis we look at the iteration (7.1), while the actual implementation follows closely Algorithm 7.1.

From (7.1) we have

$$\mathbf{x}^{(k)} = \begin{pmatrix} x_1^{(k)} \\ \mathbf{x}_2^{(k)} \end{pmatrix} = \begin{bmatrix} \lambda_1 & \mathbf{0}^* \\ \mathbf{0} & J_2 \end{bmatrix} \begin{pmatrix} x_1^{(k-1)} \\ \mathbf{x}_2^{(k-1)} \end{pmatrix} = \begin{bmatrix} \lambda_1 & \mathbf{0}^* \\ \mathbf{0} & J_2 \end{bmatrix}^k \begin{pmatrix} x_1^{(0)} \\ \mathbf{x}_2^{(0)} \end{pmatrix}.$$

Defining

$$\mathbf{y}^{(k)} := \frac{1}{\lambda_1^k}\mathbf{x}^{(k)} \tag{7.11}$$

we have

$$\mathbf{y}^{(k)} = \begin{bmatrix} 1 & \mathbf{0}^* \\ \mathbf{0} & \frac{1}{\lambda_1}J_2 \end{bmatrix} \mathbf{y}^{(k-1)}.$$

Let us assume that $y_1^{(0)} = 1$. Then $y_1^{(k)} = 1$ for all k and

$$\mathbf{y}_2^{(k)} = \frac{1}{\lambda_1}J_2\mathbf{y}_2^{(k-1)}, \qquad \frac{1}{\lambda_1}J_2 = \begin{bmatrix} \mu_2 & * & & & \\ & \mu_3 & * & & \\ & & \ddots & \ddots & \\ & & & \mu_{n-1} & * \\ & & & & \mu_n \end{bmatrix}, \qquad |\mu_k| = \frac{|\lambda_k|}{|\lambda_1|} < 1.$$

For the sequel we need

Theorem 7.3 *Let* $|||\cdot|||$ *be* any *matrix norm. Then*

$$\lim_{k\to\infty} |||M^k|||^{1/k} = \rho(M) = \max_i |\lambda_i(M)|. \tag{7.12}$$

Proof. See Horn-Johnson [3], pp.297-299. ∎

Definition 7.4 $\rho(M)$ in (7.12) is call **spectral radius** of M.

Regarding the convergence of the vector iteration, Theorem 7.3 implies that for *any* $\varepsilon > 0$ there is an integer $K(\varepsilon)$ such that

$$|||M^k|||^{1/k} \le \rho(M) + \varepsilon, \qquad \text{for all } k > K(\varepsilon). \tag{7.13}$$

We will apply this theorem to the case $M = \lambda_1^{-1}J_2$, the matrix norm $|||\cdot|||$ will be the ordinary 2-norm. Thus, for any $\varepsilon > 0$ there is a $K(\varepsilon) \in \mathbb{N}$ with

$$\left\| \left(\frac{1}{\lambda_1}J_2\right)^k \right\|^{1/k} \le |\mu_2| + \varepsilon, \quad \forall\, k > K(\varepsilon). \tag{7.14}$$

We can choose ε such that

$$|\mu_2| + \varepsilon < 1.$$

Then,

$$\sin(\angle(\mathbf{y}^{(k)}, \mathbf{e}_1)) = \frac{\|(I - \mathbf{e}_1\mathbf{e}_1^*)\mathbf{y}^{(k)}\|}{\|\mathbf{y}^{(k)}\|} = \frac{\|\mathbf{y}_2^{(k)}\|}{\|\mathbf{y}^{(k)}\|} = \frac{\|\mathbf{y}_2^{(k)}\|}{\sqrt{1 + \|\mathbf{y}_2^{(k)}\|}}$$
$$\le \|\mathbf{y}_2^{(k)}\| \le \|\frac{1}{\lambda_1^k}J_2^k\|\|\mathbf{y}_2^{(0)}\| \le (|\mu_2| + \varepsilon)^k\|\mathbf{y}_2^{(0)}\|.$$

Thus, the angle between $\mathbf{y}^{(k)}$ and $\mathbf{e}_1$ goes to zero with a rate $\mu_2 + \varepsilon$ for any positive ε. Since $\mathbf{x}^{(k)}$ is a scalar multiple of $\mathbf{y}^{(k)}$ the same holds for the angle between $\mathbf{x}^{(k)}$ and $\mathbf{e}_1$. Since we can choose ε arbitrarily small, we have proved that

$$\sin\vartheta^{(k)} = \sin(\angle(\mathbf{x}^{(k)}, \mathbf{u}_1)) \le c \cdot \left|\frac{\lambda_2}{\lambda_1}\right|^k$$

provided that $x_1^{(0)} = \mathbf{e}_1^* \mathbf{x}^{(0)} \neq 0$.

Returning to a general matrix $A \in \mathbb{F}^{n\times n}$ with Jordan normal form $A = XJY^*$, we employ equation (7.3). The sequence $\mathbf{y}^{(k)} = Y^*\mathbf{x}^{(k)}$ converges to $\mathbf{y}_* = \alpha\mathbf{e}_1$ with $\alpha \neq 0$. Therefore, $\mathbf{x}^{(k)}$ converges to a multiple of $X\mathbf{e}_1$, which is an eigenvector associated with the largest eigenvalue λ_1. The condition $\mathbf{e}_1^*\mathbf{y}^{(0)} \neq 0$ translates into

$$\mathbf{e}_1^*(Y^*\mathbf{x}^{(0)}) = (Y\mathbf{e}_1)^*\mathbf{x}^{(0)} \neq 0.$$

The first column of Y is a *left* eigenvector associated with λ_1. Therefore, we have proved

Theorem 7.5 *Let the eigenvalues of $A \in \mathbb{F}^{n\times n}$ be arranged such that $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$. Let $\mathbf{u}_1$ and $\mathbf{v}_1$ be right and left eigenvectors of A corresponding to λ_1, respectively. Then, the vector sequence generated by Algorithm 7.1 converges to $\mathbf{u}_1$ in the sense that*

$$\sin\vartheta^{(k)} = \sin(\angle(\mathbf{x}^{(k)}, \mathbf{u}_1)) \leq c \cdot \left|\frac{\lambda_2}{\lambda_1}\right|^k \tag{7.15}$$

provided that $\mathbf{v}_1^*\mathbf{x}^{(0)} \neq 0$. ■

Remark 7.1. The quantity μ_k in Algorithm 7.1 converges to $|\lambda_1|$. The true value $\lambda_1 \in \mathbb{C}$ can be found by comparing single components of $\mathbf{y}^{(k)}$ and $\mathbf{x}^{(k-1)}$. If $\lambda_1 \in \mathbb{R}$ then only the sign of λ_1 is at stake. □

Remark 7.2. The convergence of the vector iteration is faster the smaller the quotient $|\lambda_2|/|\lambda_1|$ is. □

Remark 7.3. From (7.12) we see that the norm of the powers of a matrix goes to zero if all

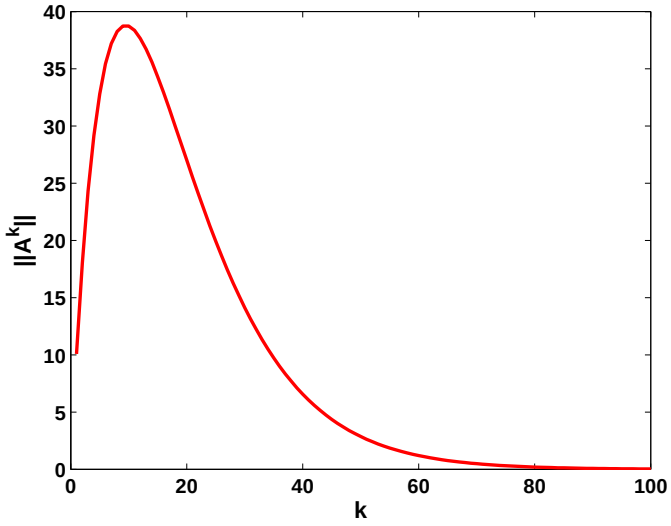

Figure 7.2: Norms of powers of B in (7.16).

is eigenvalues are smaller than one in modulus. For small powers the norm can initially grow considerably. In Fig. 7.2 we have plotted the norms of B^k with

$$B = \begin{bmatrix} 0.9 & 5 \\ 0 & 0.9 \end{bmatrix}. \tag{7.16}$$

□

Remark 7.4. If $\mathbf{v}_1^* \mathbf{x}^{(0)} = 0$ then the vector iteration converges to an eigenvector corresponding to the second largest eigenvalue. In practice, rounding errors usually prevent this behaviour of the algorithm. After a long initial phase the $\mathbf{x}^{(k)}$ turn to $\mathbf{u}_1$. □

Remark 7.5. In case that $\lambda_1 \neq \lambda_2$ but $|\lambda_1| = |\lambda_2|$ there may be no convergence at all. An example is

$$A = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad \mathbf{x}^{(0)} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

□

7.4 A numerical example

In the following MATLAB script we assume that A is upper triangular and that the largest eigenvalue (in modulus) is at position (1,1), i.e., $|a_{1,1}| > |a_{j,j}|$ for all $j > 1$.

```
%Demo  Simple Vector Iteration
%
n = 6;
randn('state',0);
A = diag([n:-1:1]) + triu(randn(n),1) % upper triangular matrix
x0 = rand(n,1); x0=x0/norm(x0)        % initial vector

e = eye(n,1); % Right eigenvector corresponding to largest
              % eigenvalue A(1,1)
% ------------------------------------------------------------

x=x0;  ang = norm(x - e*(e'*x))

hist = [ang,nan,nan];

if ~exist('tol'), tol = 100*eps; end
oldang = nan;

while ang > tol
    x = A*x;
    mue = norm(x);           % This is an approximation for the
    x = x/mue;               % searched eigenvalue

    ang = norm(x - e*(e'*x));
    hist = [hist; [mue,ang,ang/oldang]];
    oldang = ang;
end
```

Because the solution is known, we can compute the angle between iterate and true solution. We can even use this angle in the stopping criterion. The matrix A is given by

```
A =
    6.0000    1.1892   -0.5883   -0.0956   -0.6918   -0.3999
         0    5.0000    2.1832   -0.8323    0.8580    0.6900
         0         0    4.0000    0.2944    1.2540    0.8156
```

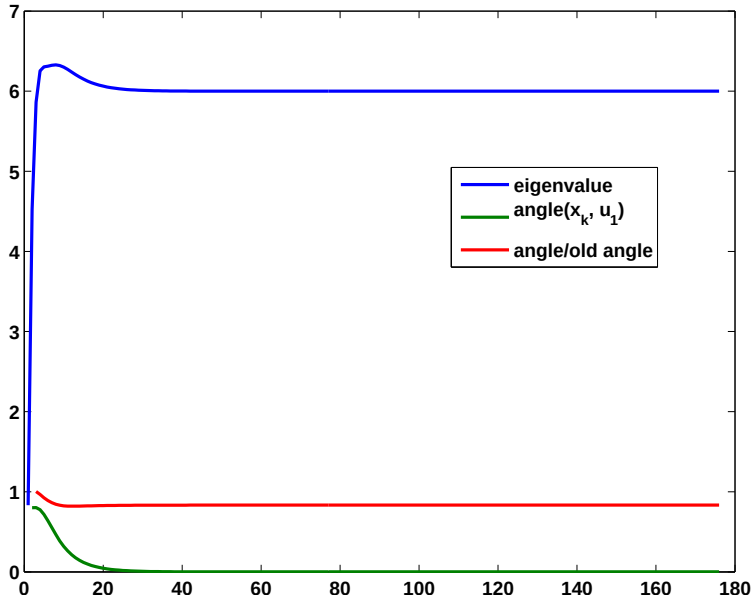

Figure 7.3: Plot of three important quantities: eigenvalue, angle between eigenvector approximation and exact eigenvector, convergence rate of eigenvector

0	0	0	3.0000	-1.5937	0.7119
0	0	0	0	2.0000	1.2902
0	0	0	0	0	1.0000

The development of three important quantities is given in Fig. 7.3. In Fig. 7.4 the case is depicted when the initial vector is chosen orthogonal to the left eigenvector corresponding to $\lambda_1 = 6$. Initially, the approximated eigenvalue is 5. Because the stopping criterion does not hold, the iteration continues until eventually rounding errors take effect.

7.5 The symmetric case

Let us now consider the Hermitian/symmetric case. We again assume the now real eigenvalues to be arranged as in (7.10). But now the Schur decomposition of A becomes its spectral decomposition,

$$A = U\Lambda U^*, \qquad U = [\mathbf{u}_1, \ldots, \mathbf{u}_n], \quad \Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n). \tag{7.17}$$

For the convergence analysis, we assume that A is diagonal, and that

$$\lambda_1 > \lambda_2 \geq \cdots \geq \lambda_n \geq 0. \tag{7.18}$$

Therefore, in (7.9) we have $J_2 = \operatorname{diag}(\lambda_2, \ldots, \lambda_n)$, i.e., all Jordan blocks are 1×1.

In contrast to the general case, in the Hermitian case we approximate the eigenvalue by the Rayleigh quotient of $\mathbf{x}^{(k)}$,

$$\lambda^{(k)} := \mathbf{x}^{(k)*} A \mathbf{x}^{(k)}, \qquad \|\mathbf{x}^{(k)}\| = 1. \tag{7.19}$$

The symmetric algorithm is given in Algorithm 7.2.

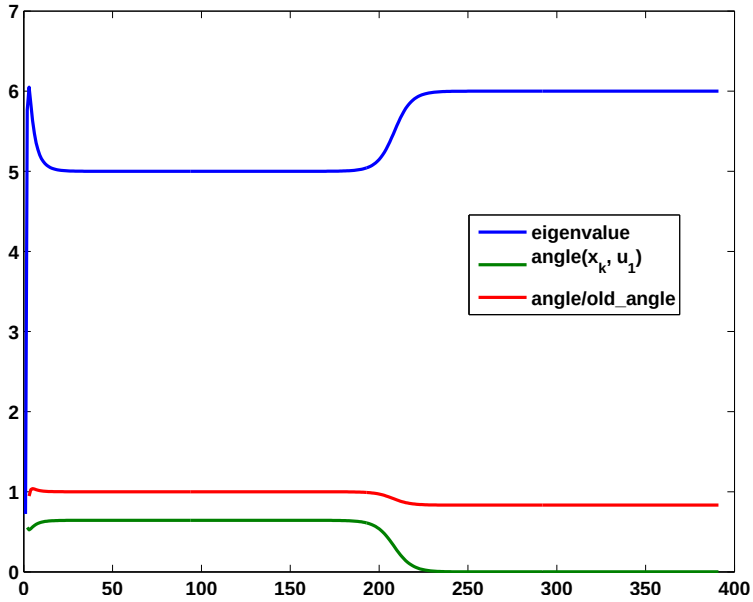

Figure 7.4: Plot of three important quantities: eigenvalue, angle between eigenvector approximation and exact eigenvector, convergence rate of eigenvector. Here, the initial vector is chosen orthogonal to the left eigenvector corresponding to the largest eigenvalue

Algorithm 7.2 Simple vector iteration for Hermitian matrices

Choose a starting vector $\mathbf{x}^{(0)} \in \mathbb{F}^n$ with $\|\mathbf{x}^{(0)}\| = 1$.
$\mathbf{y}^{(0)} := A\mathbf{x}^{(0)}$.
$\lambda^{(0)} := \mathbf{y}^{(0)*}\mathbf{x}^{(0)}$.
$k := 0$.
while $\|\mathbf{y}^{(k)} - \lambda^{(k)}\mathbf{x}^{(k)}\| > \text{tol}$ **do**
  $k := k + 1$;
  $\mathbf{x}^{(k)} := \mathbf{y}_{k-1}/\|\mathbf{y}_{k-1}\|$;
  $\mathbf{y}^{(k)} := A\mathbf{x}^{(k)}$;
  $\lambda^{(k)} := \mathbf{y}^{(k)*}\mathbf{x}^{(k)}$;
end while

In order to investigate the convergence of the Rayleigh quotient we work with auxiliary vectors

$$\mathbf{y}^{(k)} = \begin{pmatrix} 1 \\ \mathbf{y}_2^{(k)} \end{pmatrix} = \frac{1}{|x_1^{(k)}|}\mathbf{x}^{(k)}. \tag{7.20}$$

Notice, that any 'reasonable' approximation of the first eigenvector $\mathbf{e}_1$ has a nonzero first component. For the Rayleigh quotients we have

$$\rho(\mathbf{y}^{(k)}) = \rho(\mathbf{x}^{(k)}).$$

Now,

$$\lambda^{(k)} = \frac{\mathbf{y}^{(k)*}A\mathbf{y}^{(k)}}{\mathbf{y}^{(k)*}\mathbf{y}^{(k)}} = \frac{(\mathbf{e}_1 + \mathbf{y}_2^{(k)})^* A(\mathbf{e}_1 + \mathbf{y}_2^{(k)})}{1 + \|\mathbf{y}_2^{(k)}\|^2} = \frac{\lambda_1 + \mathbf{y}_2^{(k)*}A\mathbf{y}_2^{(k)}}{1 + \|\mathbf{y}_2^{(k)}\|^2} \tag{7.21}$$

where we used that $\mathbf{e}_1^* \mathbf{y}_2^{(k)} = 0$ and $\mathbf{e}_1^* A \mathbf{y}_2^{(k)} = 0$. Because,

$$\tan \vartheta^{(k)} := \tan(\angle(\mathbf{y}^{(k)}, \mathbf{e}_1)) = \|\mathbf{y}_2^{(k)}\|$$

and

$$1 + \tan^2(\phi) = \frac{1}{1 - \sin^2(\phi)}$$

we get from (7.21) that

$$\lambda^{(k)} = (\lambda_1 + \mathbf{y}_2^{(k)^*} A \mathbf{y}_2^{(k)})(1 - \sin^2 \vartheta^{(k)}) = \lambda_1 - \lambda_1 \sin^2 \vartheta^{(k)} + \mathbf{y}_2^{(k)^*} A \mathbf{y}_2^{(k)} \cos^2 \vartheta^{(k)}. \tag{7.22}$$

Now, since $\lambda_1 > 0$,

$$\begin{aligned} 0 \le \lambda_1 - \lambda^{(k)} &= \lambda_1 \sin^2 \vartheta^{(k)} - \mathbf{y}_2^{(k)^*} A \mathbf{y}_2^{(k)} \cos^2 \vartheta^{(k)} \\ &\le \lambda_1 \sin^2 \vartheta^{(k)} - \lambda_n \|\mathbf{y}_2^{(k)}\|^2 \cos^2 \vartheta^{(k)} = (\lambda_1 - \lambda_n) \sin^2 \vartheta^{(k)}. \end{aligned} \tag{7.23}$$

In summary, we have proved

Theorem 7.6 *Let A be a symmetric matrix with spectral decomposition (7.17)–(7.18). Then, the simple vector iteration of Algorithm 7.2 computes sequences $\{\lambda^{(k)}\}_{k=0}^{\infty}$ and $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ that converge linearly towards the largest eigenvalue λ_1 of A and the corresponding eigenvector $\mathbf{u}_1$ provided that the initial vector $\mathbf{x}^{(0)}$ has a nonzero component in the direction of $\mathbf{u}_1$, i.e., that $\mathbf{u}_1^* \mathbf{x}^{(0)} \neq 0$. The convergence rates are given by*

$$\sin \vartheta^{(k)} \le \left|\frac{\lambda_2}{\lambda_1}\right|^k \sin \vartheta^{(0)}, \qquad |\lambda_1 - \lambda^{(k)}| \le (\lambda_1 - \lambda_n) \left|\frac{\lambda_2}{\lambda_1}\right|^{2k} \sin^2 \vartheta^{(0)}.$$

where $\vartheta^{(k)} = \angle(\mathbf{x}^{(k)}, \mathbf{u}_1$. ■

Thus, the speed of convergence is determined by the ratio of the two eigenvalues largest in modulus and the quality of the initial guess $\mathbf{x}^{(0)}$. Both sequences $\{\lambda^{(k)}\}$ and $\{\mathbf{x}^{(k)}\}$ converge linearly, but the decisive ratio appears squared in the bound for the approximation error in the eigenvalue. $\lambda_1 - \lambda_n$ is called the **spread** of the spectrum of A. Its occurance in the bound for $\lambda_{\max} - \lambda^{(k)}$ shows that a simple scaling of the matrix does not affect the convergence behavior of the algorithm.

Example 7.7 Let's compute the *smallest* eigenvalue and corresponding eigenvector of the one-dimensional Poisson matrix $T = T_n$ of Example 2.7 with $n = 40$. Let us assume that we know an upper bound τ for the largest eigenvalue λ_n of T then the transformed matrix $\tau I - T$ has the same eigenvectors as T and eigenvalues $\tau - \lambda_n < \tau - \lambda_{n-1} < \cdots < \tau - \lambda_1$. So, we apply vector iteration to compute the desired quantities.

We set $\tau = 4(n+1)^2/\pi^2$ a number that is easily obtained by applying Gerschgorin's circle theorem. We performed a MATLAB experiment starting with a random vector.

```
>> n=40;
>> T = (4*((n+1)^2/pi^2))*eye(n) - ((n+1)^2/pi^2)*p_1d(n);
>> rand('state',0); x0=rand(n,1);
>> [x,lam,nit]=vit(T,x0,1e-4);
>> tau-lam
ans =
    0.9995
>> nit
nit =
        1968
```

In as many as 1968 iteration steps we arrived at an eigenvalue approximation 0.9995. This number is correct to all digits. The difference to the eigenvalue 1 of the continuous eigenvalue problem $-u''(x) = \lambda u(x)$ is due to the discretization error. Figure 7.5 shows the convergence history of this calculation. The straight lines show the actual angle $\vartheta^{(k)}$

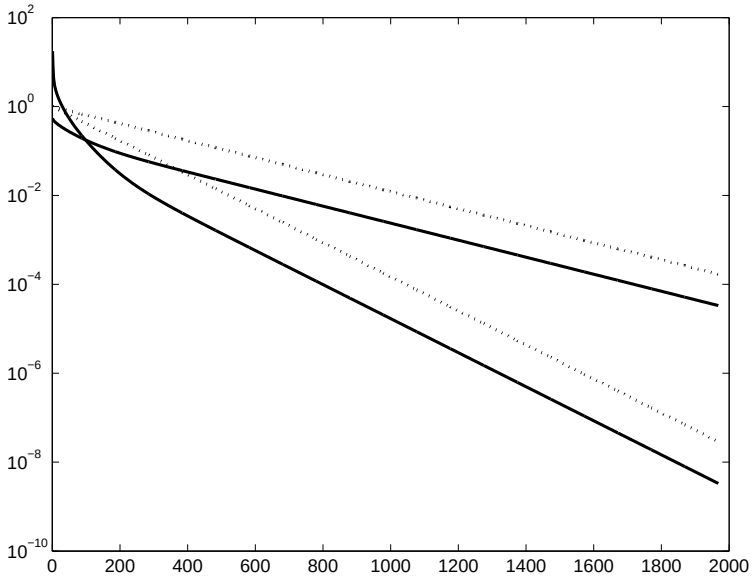

Figure 7.5: Simple vector iteration with $\tau I_{40} - T_{40}$

between $\mathbf{x}^{(k)}$ and $\mathbf{u}_1$ (above) and the actual error $\lambda^{(k)} - \lambda_1$. These quantities can of course not be computed in general. In this example we know them, see Ex. 2.7. The dotted lines show powers of $q = (\tau - \lambda_2)/(\tau - \lambda_1)$ that indicate the convergence rates given by Theorem 7.6. Here, $q = 0.9956$. Clearly, the convergence is as predicted.

Example 7.8 We mentioned that a good initial vector can reduce the number of iteration steps. Remember that the smallest eigenfunction is $\sin x$, a function that is positive on the whole interval $(0, \pi)$. Let us therefore set $\mathbf{x}^{(0)}$ to be the vector of all ones.

```
>> x0 = ones(n,1);
>> [x,lam,nit]=vit(T,x0,1e-4);
>> nit
nit =
   866
```

This is a surprisingly high reduction in the number of iteration steps. Figure 7.6 shows the convergence history of this calculation. Again the doted lines indicate the convergence rates that are to be expected. The actual convergence rates are evidently much better. How can that be?

The eigenvectors of T_n resemble the eigenfunctions $\sin kx$ of the continuous eigenvalue problem. Therefore the coefficients corresponding to eigenvectors corresponding to eigenfunctions antisymmetric with respect to the point $\pi/2$ vanish. In particular $x_2 = 0$. Therefore the convergence rate is not $q = (\tau - \lambda_2)/(\tau - \lambda_1)$ but $\hat{q} = (\tau - \lambda_3)/(\tau - \lambda_1)$. This is verified by the numbers given in Fig. 7.7 where the assymptotic corrected convergence rates $\hat{q}$ and $\hat{q}^2$ are indicated.

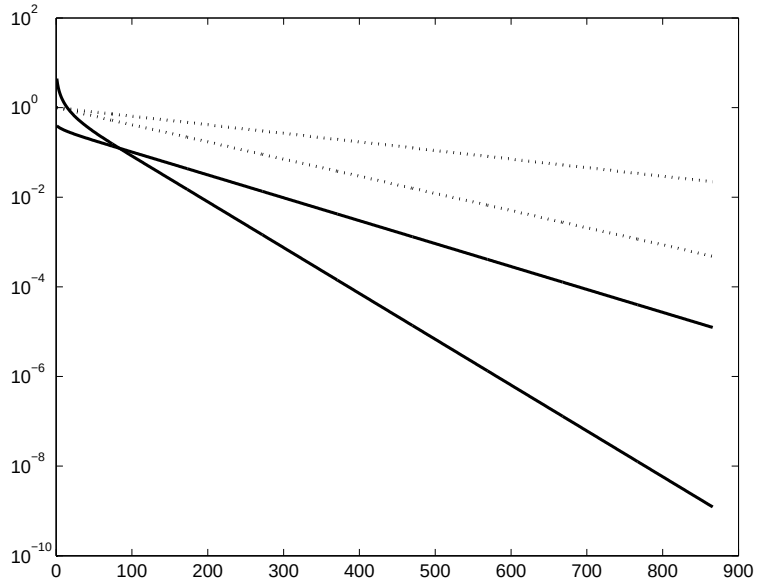

Figure 7.6: Simple vector iteration with $\tau I_{40} - T_{40}$ and starting vector $(1, 1, \ldots, 1)^T$

Problem 7.9 When computing the smallest eigenvalue of T_n by the simple vector iteration we can find a better shift than τ above if the extremal points of the spectrum are known. Determine σ such that $\sigma I_n - T_n$ exhibits the optimal convergence rate. Hint: On the one hand we would like the quotient $(\sigma - \lambda_{n-1})/(\sigma - \lambda_n)$ to be as small as possible. On the other hand $|\sigma - \lambda_1|/(\sigma - \lambda_n)$ must not become to big. Hint: Equate the two quantities.

7.6 Inverse vector iteration

The previous examples have shown that the convergence of simple vector iteration is potentially very slow. The quotient of the second largest to the largest eigenvalue are very close to 1. We noticed this by means of a very simple and small eigenvalue problem. The situation gets much worse if the problems are big.

We have seen in (2.28) that a polynomial in A has the same eigenvectors as A. We therefore may try to find a polynomial that enhances the eigenvalue that we are looking for. This approach is however not successful in the most critical case when the wanted eigenvalue is very close to unwanted. In this situation, the **shift-and-invert** spectral transformation is most appropriate. Instead of a polynomial we transform the matrix by the rational function $f(\lambda) = 1/(\lambda - \sigma)$ where σ is a so-called **shift** that is chosen close to the desired eigenvalue. Simple vector iteration with the matrix $(A - \sigma I)^{-1}$ is referred to as **inverse vector iteration**, see Algorithm 7.6.

$$\mathbf{x}^{(k)} := (A - \sigma I)^{-1}\mathbf{x}^{(k-1)} \quad \Longleftrightarrow \quad (A - \sigma I)\,\mathbf{x}^{(k)} := \mathbf{x}^{(k-1)}, \qquad k = 1, 2, \ldots \tag{7.24}$$

The iteration converges towards the eigenvector with eigenvalue closest to σ. A linear system of equations has to be solved in each iteration step. Of course only one Cholesky or LU factorization has to be computed as the shift remains constants in all iterations. The stopping criterion is changed into

$$\|\mathbf{x}^{(k)} - \mathbf{y}^{(k)}/\mu^{(k)}\| \leq \mathrm{tol}\|\mathbf{y}^{(k)}\| \tag{7.25}$$

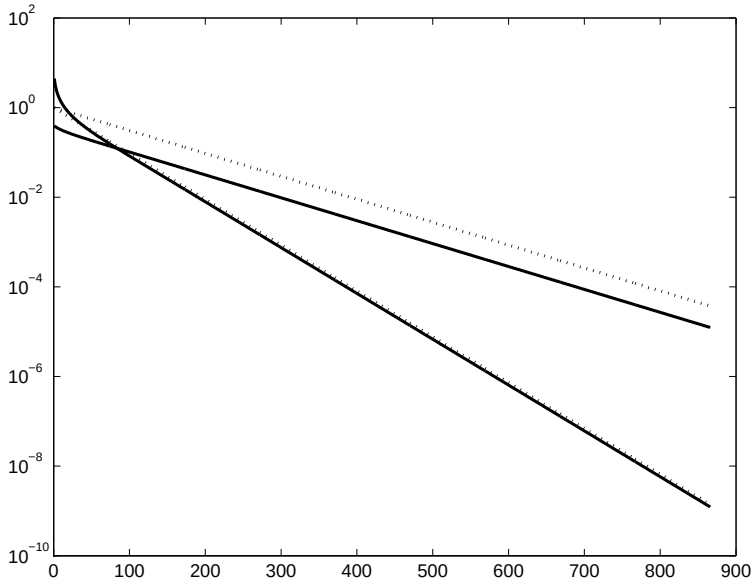

Figure 7.7: Simple vector iteration with $\tau I_{40} - T_{40}$ and starting vector $(1, 1, \ldots, 1)^T$

Algorithm 7.3 Inverse vector iteration

1: Choose a starting vector $\mathbf{x}_0 \in \mathbb{F}^n$ and a shift σ.
2: Compute the LU factorization of $A - \sigma I$: $LU = P(A - \sigma I)$
3: $\mathbf{y}^{(0)} := U^{-1}L^{-1}P\mathbf{x}^{(0)}$. $\mu^{(0)} = \mathbf{y}^{(0)*}\mathbf{x}^{(0)}$, $\lambda^{(0)} := \sigma + 1/\mu^{(0)}$. $k := 0$.
4: **while** $\|\mathbf{x}^{(k)} - \mathbf{y}^{(k)}/\mu^{(k)}\| > \mathrm{tol}\|\mathbf{y}^{(k)}\|$ **do**
5: $k := k + 1$.
6: $\mathbf{x}^{(k)} := \mathbf{y}_{k-1}/\|\mathbf{y}_{k-1}\|$.
7: $\mathbf{y}^{(k)} := U^{-1}L^{-1}P\mathbf{x}^{(k)}$.
8: $\mu^{(k)} := \mathbf{y}^{(k)*}\mathbf{x}^{(k)}$, $\lambda^{(k)} := \sigma + 1/\mu^{(k)}$.
9: **end while**

where we have used

$$A\mathbf{y}^{(k)} - \lambda^{(k)}\mathbf{y}^{(k)} = A\mathbf{y}^{(k)} - \left(\sigma - \frac{1}{\mu^{(k)}}\right)\mathbf{y}^{(k)} = \mathbf{x}^{(k)} - \mathbf{y}^{(k)}/\mu^{(k)}$$

The convergence result of Theorem 7.6 can easily be adapted to the new situation if it is taken into account that $A - \sigma I$ has eigenpairs $(\mu_i, \mathbf{u}_i)$ with $\mu_i = 1/(\sigma - \lambda_i)$.

Theorem 7.10 *Let A be symmetric positive definite with spectral decomposition (7.17). Let $\lambda'_1, \ldots, \lambda'_n$ be a renumeration of the eigenvalues in (7.17) such that*

$$\frac{1}{|\lambda'_1 - \sigma|} > \frac{1}{|\lambda'_2 - \sigma|} \geq \cdots \geq \frac{1}{|\lambda'_n - \sigma|} \tag{7.26}$$

Then, provided that $\mathbf{u}'^{*}_1\mathbf{x}^{(0)} \neq 0$, the inverse vector iteration of Algorithm 7.6 constructs sequences $\{\lambda^{(k)}\}_{k=0}^{\infty}$ and $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ that converge linearly towards that eigenvalue λ'_1 closest to the shift σ and to the corresponding eigenvector $\mathbf{u}'_1$, respectively. The bounds

$$\sin\vartheta^{(k)} \leq \left|\frac{\lambda'_1 - \sigma}{\lambda'_2 - \sigma}\right|^k \sin\vartheta^{(0)}, \qquad \lambda^{(k)} - \lambda_1 \leq \delta\left|\frac{\lambda'_1 - \sigma}{\lambda'_2 - \sigma}\right|^{2k} \sin^2\vartheta^{(0)}.$$

hold with $\vartheta^{(k)} = \angle(\mathbf{x}^{(k)}, \mathbf{u}_1)$ and $\delta = \mathrm{spread}(\sigma((A - \sigma I)^{-1}))$. ■

If the shift σ approximates very well an eigenvalue of A then $\frac{\lambda^{(k)}-\sigma}{\lambda_n-\sigma} \ll 1$ and convergence is very rapid.

Example 7.11 Let us now use inverse iteration to compute the smallest eigenvalue and corresponding eigenvector of the one-dimensional Poisson matrix $T = T_n$ of Example 2.7 with $n = 40$. If we assume that we know that the smallest eigenvalue λ_1 is around 1 then a shift $\sigma = .9$ is reasonable, if we want $A - \sigma I$ to still be positive definite. Starting with the vector of all ones three iteration steps suffice to get the desired accuracy of tol $= 10^{-5}$, see Table 7.1.

k	$\lambda^{(k)} - \lambda_1$	$\sin(\vartheta^{(k)})$
1	2.0188e-02	4.1954e-03
2	1.7306e-06	5.0727e-05
3	2.5289e-10	6.2492e-07

Table 7.1: Computing the lowest eigenvalue of the one-dimensinal Poisson equation by inverse iteration

Example 7.12 We consider the problem of computing the eigenvector corresponding to a *known* eigenvalue. The matrix that we consider is one of the so-called Wilkinson matrices

$$T = \begin{bmatrix} 19 & -1 & & & & & & & \\ -1 & 18 & -1 & & & & & & \\ & \ddots & \ddots & \ddots & & & & & \\ & & -1 & 1 & -1 & & & & \\ & & & -1 & 1 & -1 & & & \\ & & & & \ddots & \ddots & \ddots & & \\ & & & & & -1 & 19 & -1 & \\ & & & & & & -1 & 20 \end{bmatrix}.$$

Wilkinson matrices are irreducible tridiagonal matrices that have very close eigenvalues. This matrix has the eigenvalues

i	λ_i	i	λ_i
1	-1.1254415221199814	11	5.0002362656192743
2	0.2538058170966502	12	5.9999918413270530
3	0.9475343675285830	13	6.0000083521880692
4	1.7893213526669509	14	6.9999997949295611
5	2.1302092192694015	15	7.0000002079042920
6	2.9610588806935558	16	7.9999999961918720
7	3.0430992883895192	17	8.0000000038418246
8	3.9960479973346419	18	8.9999999999455120
9	4.0043538173235769	19	9.0000000000548166
10	4.9997743198148310	20	9.9999999999996234

The following MATLAB code constructs the SPARSE tridiagonal matrix T.

```
n = 40;
e = ones(n,1); f = abs([-n/2+1:n/2]');
T = spdiags([-e f -e], [-1:1], n, n);
lam = sort(eig(T));
```

Computing the 20-th and 21-st eigenvectors could be done in the following way.

```
>> x = (T - lam(20)*eye(n))\e;
>> y = (T - lam(21)*eye(n))\e;
>> x = x/norm(x);  y = y/norm(y);
>> x'*y
ans =
   0.00140329005834
>> norm((T - lam(20)*eye(n))*x)
ans =
     7.325760095786749e-15
>> norm((T - lam(21)*eye(n))*y)
ans =
     7.120036319503636e-15
```

The computed vectors $\mathbf{x}$ and $\mathbf{y}$ are good approximations in the sense that they give small residuals. However, the two vectors are nor mutually orthogonal at all. We try to improve orthogonality by apply a second step of inverse iteration

```
>> x = (T - lam(20)*eye(n))\x;
>> y = (T - lam(21)*eye(n))\y;
>> x = x/norm(x);  y = y/norm(y);
>> x'*y
ans =
    -1.313592004487587e-05
```

Things have only slightly improved. Therefore, we orthogonalize $\mathbf{y}$ *explicitely* against $\mathbf{x}$.

```
>> y = y - x*(x'*y);
>> x'*y
ans =
    -2.155571068436496e-17
>> norm((T - lam(21)*eye(n))*y)
ans =
     1.557058217172078e-15
>> norm((T - lam(20)*eye(n))*x)
ans =
     4.117116818055497e-16
```

This helped. The two eigenvectors are now perpendicular on each other, and the residuals are still fine.

Discussion of inverse iteration

We have seen that

- we can compute eigenvectors corresponding to any (simple and well separated) eigenvalue if we choose the shift properly, and that
- we have very good convergence rates, is the shift is close to an eigenvalue.

However, one may feel uncomfortable solving an 'almost singular' system of equations, after all $\sigma \approx \lambda_k$ means that the condition of $A-\sigma I$ is very big. From the analysis of linear systems of equations we know that this means large errors in the solution. Furtunately, the error that is suffered from when solving with $A - \sigma I$ points in the right direction. To see this, assume that the singular value decomposition of $A - \sigma I$ is given by

$$A - \sigma I = U\Sigma V^*, \qquad \Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n), \quad \text{with } \sigma_1 \geq \cdots \geq \sigma_n \geq 0.$$

If $A - \sigma I$ is 'almost' singular then $\sigma_n \ll 1$. If even $\sigma_n = 0$ then $(A - \sigma I)\mathbf{v}_n = \mathbf{0}$, i.e., the last right singular vector is an eigenvector of A corresponding to the eigenvalue σ (the shift).

If $\sigma_n = \mathcal{O}(\varepsilon)$ then

$$(A - \sigma I)\mathbf{z} = U\Sigma V^*\mathbf{z} = \mathbf{y}.$$

Thus,

$$\mathbf{z} = V\Sigma^{-1}U^*\mathbf{y} = \sum_{i=1}^{n} \mathbf{v}_i \frac{\mathbf{u}_i^*\mathbf{y}}{\sigma_i} \overset{\sigma_n \ll \sigma_{n-1}}{\approx} \mathbf{v}_n \frac{\mathbf{u}_n^*\mathbf{y}}{\sigma_n}.$$

The tiny σ_n blows up the component in direction of $\mathbf{v}_n$. So, the vector $\mathbf{z}$ points in the desired 'most singular' direction.

7.7 The generalized eigenvalue problem

Applying the vector iteration (7.1) to the generalized eigenvalue problem $A\mathbf{x} = \lambda B\mathbf{x}$ leads to the iteration

$$\mathbf{x}^{(k)} := B^{-1}A\mathbf{x}^{(k-1)}, \qquad k = 1, 2, \ldots$$

Since the solution of a linear system is required in each iteration step, we can execute an inverse iteration right-away,

$$(A - \sigma B)\,\mathbf{x}^{(k)} := B\mathbf{x}^{(k-1)}, \qquad k = 1, 2, \ldots \tag{7.27}$$

The iteration performs an ordinary vector iteration for the eigenvalue problem

$$(A - \sigma B)^{-1}B\mathbf{x} := \mu\mathbf{x}, \qquad \mu = \frac{1}{\lambda - \sigma}. \tag{7.28}$$

Thus, the iteration (7.27) converges to the largest eigenvector of (7.28), i.e., the eigenvector with eigenvalue closest to the shift σ.

7.8 Computing higher eigenvalues

In order to compute higher eigenvalues $\lambda_2, \lambda_3, \ldots$, we make use of the mutual orthogonality of the eigenvectors of symmetric matrices, see Theorem 2.14. (In the case of Schur vectors we can proceed in a similar way.)

So, in order to be able to compute the second eigenpair $(\lambda_2, \mathbf{u}_2)$ we have to know the eigenvector $\mathbf{u}_1$ corresponding to the lowest eigenvalue. Most probably is has been computed previously. If this is the case we can execute an inverse iteration orthogonal to $\mathbf{u}_1$.

More generally, we can compute the j-th eigenpair $(\lambda_j, \mathbf{u}_j)$ by inverse iteration, keeping the iterated vector $\mathbf{x}^{(k)}$ orthogonal to the already known or computed eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_{j-1}$.

Algorithm 7.4 Inverse vector iteration for computing $(\lambda_j, \mathbf{u}_j)$

The LU factorization of $A - \sigma I$: $LU = P(A - \sigma I)$
and the eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_{j-1}$ are known.
Choose a starting vector $\mathbf{x}^{(0)}$ such that $\mathbf{u}_q^* \mathbf{x}^{(0)} = 0$, $q < j$.
Set $k := 0$.
while $\|\mathbf{x}^{(k)} - \mathbf{y}^{(k)}/\mu^{(k)}\| > \mathrm{tol}\|\mathbf{y}^{(k)}\|$ **do**
  $k := k + 1$;
  $\mathbf{x}^{(k)} := \mathbf{y}^{(k-1)}/\|\mathbf{y}^{(k-1)}\|$;
  $\mathbf{y}^{(k)} := U^{-1}L^{-1}P\mathbf{x}^{(k)}$;
  $\mu^{(k)} := {\mathbf{y}^{(k)}}^* \mathbf{x}^{(k)}, \quad \lambda^{(k)} := \sigma + 1/\mu^{(k)}$.
end while

In exact arithmetic, the condition $\mathbf{u}_1^* \mathbf{x}^{(0)} = \cdots = \mathbf{u}_{j-1}^* \mathbf{x}^{(0)} = 0$ implies that all $\mathbf{x}^{(k)}$ are orthogonal to $\mathbf{u}_1, \ldots, \mathbf{u}_{j-1}$. In general, however, one has to expect rounding errors that introduce components in the directions of already computed eigenvectors. Therefore, it is *necessary* to enforce the orthogonality conditions during the iteration.

Assuming exact arithmetic, Theorem 7.10 immediately implies that

$$\sin \angle(\mathbf{x}^{(k)}, \mathbf{x}_j) \le c_1 \left(\frac{\lambda_j}{\lambda_{j'}}\right)^k$$
$$|\lambda^{(k)} - \lambda_j| \le c_2 \left(\frac{\lambda_j}{\lambda_{j'}}\right)^{2k}$$

where j' is the smallest index for which $\lambda_{j'} > \lambda_j$.

7.9 Rayleigh quotient iteration

We now assume that the matrix the eigenpairs of which we want to determine is Hermitian (or symmetric).

We have noticed that inverse iteration is an effective way to compute eigenpairs, if a good approximation of the desired eigenvalue is known. This approximation is used as a shift. However, as we have seen earlier, if a good approximation of an eigenvector is available its Rayleigh quotient gives a very good approximation of its eigenvalue.

Indeed we have the following

Lemma 7.13 *Let* $\mathbf{q}$ *be any nonzero vector. The number* ρ *that minimizes* $\|A\mathbf{q} - \rho\mathbf{q}\|$ *is the* Rayleigh quotient

$$\rho = \frac{\mathbf{q}^* A \mathbf{q}}{\mathbf{q}^* \mathbf{q}}. \tag{7.29}$$

Proof. Let $\rho \in \mathbb{R}$ be the Rayleigh quotient (7.29) of $\mathbf{q} \neq 0$ and let $\tau \in \mathbb{C}$ be any number. Then we have

$$\begin{aligned}
\|A\mathbf{q} - (\rho + \tau)\mathbf{q}\|^2 &= \mathbf{q}^* A^2 \mathbf{q} - (2\rho + \tau + \bar{\tau})\, \mathbf{q}^* A \mathbf{q} + |\rho + \tau|^2 \mathbf{q}^* \mathbf{q} \\
&= \mathbf{q}^* A^2 \mathbf{q} - 2\rho\, \mathbf{q}^* A \mathbf{q} - 2\mathrm{Re}(\tau)\, \mathbf{q}^* A \mathbf{q} + \rho^2\, \mathbf{q}^* \mathbf{q} + 2\rho \mathrm{Re}(\tau)\, \mathbf{q}^* \mathbf{q} + |\tau|^2 \mathbf{q}^* \mathbf{q} \\
&= \mathbf{q}^* A^2 \mathbf{q} - \frac{(\mathbf{q}^* A \mathbf{q})^2}{\mathbf{q}^* \mathbf{q}} + |\tau|^2\, \mathbf{q}^* \mathbf{q}.
\end{aligned}$$

The last term is smallest if $\tau = 0$. ∎

Algorithm 7.5 Rayleigh quotient iteration (RQI)

1: Choose a starting vector $\mathbf{y}_0 \in \mathbb{F}^n$ with $\|\mathbf{y}_0\| = 1$ and a tolerance ε.
2: **for** $k = 1, 2, \ldots$ **do**
3: $\quad \rho^{(k)} := \mathbf{y}^{(k-1)^*} A \mathbf{y}^{(k-1)}$.
4: $\quad$ Solve $(A - \rho^{(k)} I)\mathbf{z}^{(k)} = \mathbf{y}^{(k-1)}$ for $\mathbf{z}^{(k)}$.
5: $\quad \sigma^{(k)} = \|\mathbf{z}^{(k)}\|$.
6: $\quad \mathbf{y}^{(k)} := \mathbf{z}^{(k)}/\sigma^{(k)}$.
7: $\quad$ **if** $\sigma^{(k)} > 10/\varepsilon$ **then**
8: $\quad\quad$ **return** $\{\mathbf{y}^{(k)}\}$
9: $\quad$ **end if**
10: **end for**

The following algorithm 7.9 is a modification of inverse iteration. In each iteration step the shift is modified to be the Rayleigh quotient of the most recent eigenvector approximation. This is not a curse but a blessing [4] as we have seen in section 7.6.

In step 4 of this algorithm a close to singular system of equation is solved. This results in a very long solution whose norm is used a the convergence criterion.

The Rayleigh quotient iteration usually converges, however not always towards the desired solution. Therefore, to investigate the convergence rate we make the following

Assumption: $\mathbf{y}^{(k)} \xrightarrow[k\to\infty]{} \mathbf{x}$ with $A\mathbf{x} = \lambda\mathbf{x}$.

This assumption garantees that there is at all convergence towards a certain eigenvector $\mathbf{x}$. Let $\|\mathbf{x}\| = \|\mathbf{y}^{(k)}\| = 1$ and let the angle between this eigenvector and its approximation be $\varphi^{(k)} = \angle(\mathbf{x}, \mathbf{y}^{(k)})$. Then the assumption implies that $\{\varphi^{(k)}\}_{k=1}^{\infty}$ converges to zero. We can write

$$\mathbf{y}^{(k)} = \mathbf{x}\cos\varphi^{(k)} + \mathbf{u}^{(k)}\sin\varphi^{(k)}, \qquad \|\mathbf{x}\| = \|\mathbf{y}^{(k)}\| = \|\mathbf{u}^{(k)}\| = 1.$$

Let

$$\rho^{(k)} = \rho(\mathbf{y}^{(k)}) = \frac{\mathbf{y}^{(k)^*} A \mathbf{y}^{(k)}}{\mathbf{y}^{(k)^*}\mathbf{y}^{(k)}} = \mathbf{y}^{(k)^*} A \mathbf{y}^{(k)}$$

be the Rayleigh quotient of $\mathbf{y}_k$. Then we have

$$\begin{aligned}\lambda - \rho_k &= \lambda - \cos^2\varphi_k \underbrace{\mathbf{x}^* A \mathbf{x}}_{\lambda} - \cos\varphi_k \sin\varphi_k \underbrace{\mathbf{x}^* A \mathbf{u}_k}_{0} - \sin^2\varphi_k \mathbf{u}_k^* A \mathbf{u}_k \\ &= \lambda(1 - \cos^2\varphi_k) - \sin^2\varphi_k \rho(\mathbf{u}_k) \\ &= (\lambda - \rho(\mathbf{u}_k))\sin^2\varphi_k.\end{aligned}$$

We now prove the

Theorem 7.14 (Local convergence of Rayleigh quotient iteration) *With the above assumption we have*

$$\lim_{k\to\infty} \left| \frac{\varphi_{k+1}}{\varphi_k^3} \right| \le 1. \tag{7.30}$$

i.e., RQI converges cubically.

Proof. (The proof follows closely the one given by Parlett [4].) We have

$$\begin{aligned}\mathbf{z}_{k+1} = (A - \rho_k I)^{-1}\mathbf{y}_k &= \mathbf{x}\cos\varphi_k/(\lambda - \rho_k) + (A - \rho_k I)^{-1}\mathbf{u}_k \sin\varphi_k \\ &= \mathbf{x}\underbrace{\cos\varphi_k/(\lambda - \rho_k)}_{\|\mathbf{z_{k+1}}\|\cos\varphi_{k+1}} + \mathbf{u}_{k+1}\underbrace{\sin\varphi_k \|(A - \rho_k I)^{-1}\mathbf{u}_k\|}_{\|\mathbf{z_{k+1}}\|\sin\varphi_{k+1}},\end{aligned}$$

where we set

$$\mathbf{u}_{k+1} := (A-\rho_k I)^{-1}\mathbf{u}_k / \|(A-\rho_k I)^{-1}\mathbf{u}_k\| \tag{7.31}$$

such that $\|\mathbf{u}_{k+1}\| = 1$ and $\mathbf{u}_{k+1}^*\mathbf{x} = 0$. Thus,

$$\begin{aligned}\tan\varphi_{k+1} &= \sin\varphi_{k+1}/\cos\varphi_{k+1}\\ &= \sin\varphi_k\,\|(A-\rho_k I)^{-1}\mathbf{u}_k\|\,(\lambda-\rho_k)/\cos\varphi_k\\ &= (\lambda-\rho_k)\,\|(A-\rho_k I)^{-1}\mathbf{u}_k\|\,\tan\varphi_k\\ &= (\lambda-\rho(\mathbf{u}_k))\,\|(A-\rho_k I)^{-1}\mathbf{u}_k\|\,\sin^2\varphi_k\tan\varphi_k.\end{aligned}$$

So,

$$(A-\rho_k I)^{-1}\mathbf{u}_k = (A-\rho_k I)^{-1}\left(\sum_{\lambda_i\neq\lambda}\beta_i\mathbf{x}_i\right) = \sum_{\lambda_i\neq\lambda}\frac{\beta_i}{\lambda_i-\rho_k}\mathbf{x}_i$$

and taking norms,

$$\|(A-\rho_k I)^{-1}\mathbf{u}_k\|^2 = \sum_{\lambda_i\neq\lambda}\frac{\beta_i^2}{|\lambda_i-\rho_k|^2} \geq \frac{1}{\min_{\lambda_i\neq\lambda}|\lambda_i-\rho_k|^2}\underbrace{\sum_{\lambda_i\neq\lambda}\beta_i^2}_{\|\mathbf{u}_k\|^2=1} \tag{7.32}$$

We define the **gap** between the eigenvalue λ and the rest of A's spectrum by

$$\gamma := \min_{\lambda_i\neq\lambda}|\lambda_i-\lambda|.$$

The assumption implies that there must be a $k_0\in\mathbb{N}$ such that $|\lambda-\rho_k| < \frac{\gamma}{2}$ for all $k > k_0$, and, therefore,

$$|\lambda_i-\rho_k| > \frac{\gamma}{2} \qquad \text{for all } \lambda_i\neq\lambda.$$

Using this in (7.32) gives

$$\|(A-\rho_k I)^{-1}\mathbf{u}_k\| \leq \frac{1}{\min_{\lambda_i\neq\lambda}|\lambda_i-\rho_k|} \leq \frac{2}{\gamma}, \qquad k > k_0.$$

Because $\tan\varphi_k \approx \sin\varphi_k \approx \varphi_k$ if $\varphi_k \ll 1$ we can deduce the cubic convergence rate.

We now look at the sequence $\{\mathbf{u}_k\}$ more closely. We note from (7.31) that this sequence is obtained by "*inverse iteration with variable shift* ρ_k". But since $\rho_k \longrightarrow \lambda$ with a cubic rate of convergence we can for large k assume that $\rho_k = \lambda$ and that $\mathbf{u}_k \perp \mathbf{x}$.

We now consider two cases, either $\{\mathbf{u}_k\}$ converges, or it does not converge.

1. We assume that $\{\mathbf{u}_k\}$ converges. Then the limit vector $\hat{\mathbf{u}}$ must be an eigenvector of A in $\operatorname{span}\{\mathbf{x}\}^\perp$. (In general, $\hat{\mathbf{u}}$ is an eigenvector corresponding to the eigenvalue $\hat{\lambda}$ that is closest to λ.) Thus,

$$(\lambda-\rho(\mathbf{u}_k))\|(A-\rho_k I)^{-1}\mathbf{u}_k\| \xrightarrow[k\to\infty]{} \pm|\lambda-\hat{\lambda}|\cdot\|\hat{\mathbf{u}}/(\lambda-\hat{\lambda})\| = \pm 1.$$

2. Now we assume that $\{\mathbf{u}_k\}$ does *not* converge. Then A has two eigenvalues of equal distance to λ and the cluster points of the sequence $\{\mathbf{u}_k\}$ are two vectors in the plane

that is spanned by two eigenvectors corresponding to these two eigenvalues $\lambda \pm \delta$, $\alpha \mathbf{x}_p + \beta \mathbf{x}_q$, where $\alpha \neq 0$, $\beta \neq 0$, and $\alpha^2 + \beta^2 = 1$. Their Rayleigh quotients are

$$\rho(\alpha \mathbf{x}_p + \beta \mathbf{x}_q) = \alpha^2 \lambda_p + \beta^2 \lambda_q = \alpha^2(\lambda \pm \delta) + \beta^2(\lambda \mp \delta) = \lambda \pm \delta(\alpha^2 - \beta^2).$$

As $k \longrightarrow \infty$ the Rayleigh quotients of $\mathbf{u}_k$ jump between these two values. Hence,

$$(\lambda - \rho(\mathbf{u}_k))\|(A - \rho_k I)^{-1}\mathbf{u}_k\| \longrightarrow \pm\delta(\alpha^2 - \beta^2)/\delta,$$

and, therefore,

$$\left|\frac{\varphi_{k+1}}{\varphi_k^3}\right| \xrightarrow[k\to\infty]{} |\alpha^2 - \beta^2| < 1$$

■

Remark 7.6. Notice that we have not proved global convergence. Regarding this issue consult the book by Parlett [4] that contains all of this and more.

RQI converges 'almost always'. However, it is not clear in general towards which eigenpair the iteration converges. So, it is wise to start RQI only with good starting vectors. An alternative is to first apply inverse vector iteration and switch to Rayleigh quotient iteration as soon as the iterate is close enough to the solution. For references on this technique see [6, 1]. □

Remark 7.7. The Rayleigh quotient iteration is expensive. In every iteration step another system of equations has to be solved, i.e., in every iteration step a matrix has to be factorized. Therefore, RQI is usually applied only to tridiagonal matrices. □

7.9.1 A numerical example

The following MATLAB script demonstrates the power of Rayleigh quotient iteration. It expects as input a matrix `A`, an initial vector `x` of length one.

```
%       Initializations
k = 0; rho = 0; ynorm = 0;

while abs(rho)*ynorm < 1e+15,
   k = k + 1; if k>20, break, end
   rho = x'*A*x;
   y = (A - rho*eye(size(A)))\x;
   ynorm = norm(y);
   x = y/ynorm;
end
```

We invoked this routine with the matrix

$$A = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{9\times 9}$$

and the initial vector $\mathbf{x} = [-4, -3, \ldots, 4]^T$. The numbers obtained are

k	rho	ynorm
1	0.6666666666666666	3.1717e+00
2	0.4155307724080958	2.9314e+01
3	0.3820048793104663	2.5728e+04
4	0.3819660112501632	1.7207e+13
5	0.3819660112501051	2.6854e+16

The cubic convergence is evident.

Bibliography

[1] C. Beattie and D. Fox, *Localization criteria and containment for Rayleigh Quotient iteration*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 80–93.

[2] G. H. Golub and C. F. van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 4th ed., 2012.

[3] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.

[4] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice Hall, Englewood Cliffs, NJ, 1980. (Republished by SIAM, Philadelphia, 1998.).

[5] H. R. Schwarz, *Methode der finiten Elemente*, Teubner, Stuttgart, 3rd ed., 1991.

[6] D. B. Szyld, *Criteria for combining inverse and Rayleigh Quotient iteration*, SIAM J. Numer. Anal., 25 (1988), pp. 1369–1375.