1 Iterative methods for large sparse eigenvalue problems

The eigenvalue problem is one of the fundamental problems in science. We wish to compute pairs $\lambda \in \mathbb{C}$ and $x \in \mathbb{C}^n$ such that

$$Ax = \lambda x, \tag{1.1}$$

and $A \in \mathbb{R}^{n\times n}$. The eigenvector is in this block assumed to be normalized as $\|x\| = 1$, with Euclidean norm such that $\|x\|^2 = x^H x = 1$.

1.1 Basic methods

1.1.1 Computing eigenvalues from eigenvectors

Before diving into the algorithms for eigenvalue problems, we will treat an easier problem.
Problem: Suppose $x \in \mathbb{C}^n$ is an approximation of an eigenvector, compute an associated eigenvalue.

Assume for the moment an idealized situation where x is exactly an eigenvector. This means that (1.1) is satisfied, and we can multiply the equation from the left with x^H:

$$x^H Ax = \lambda x^H x,$$

such that

$$\lambda = \frac{x^H Ax}{x^H x}$$

This quotient can be used also if x is not an eigenvector and is usually referred to as the Rayleigh quotient.

Definition 1.1.1 (Rayleigh quotient)**.** *The quotient defined by*

$$r(x) := \frac{x^H Ax}{x^H x}$$

is referred to as the Rayleigh quotient.

For symmetric matrices, there are additional interpretations of the Rayleigh quotient. Given an approximate eigenvector x, it minimizes $Ax - \mu x$ in the Euclidean norm: One can show that

$$\underset{\mu\in\mathbb{R}}{\operatorname{argmin}}\|\mu x - Ax\| = \frac{x^H Ax}{x^H x}$$

The Rayleigh quotient will in general only give you an approximation of the eigenvalue. The propagation of the approximation error can also be precisely described if x is sufficiently close to an eigenvector. More precisely, if we suppose $x \in \mathbb{C}^n$ is an eigenvector corresponding to an eigenvalue λ, we have that

$$r(x+\varepsilon y) = \lambda + \mathcal{O}(\varepsilon). \qquad (1.2)$$

In words, the error in the eigenvalue from the Rayleigh quotient is essentially of the order of magnitude of the error in the eigenvector. In the following this is made more concrete with an example and a theorem describing the accuracy. If A is symmetric (or hermitian) we have

$$r(x+\varepsilon y) = \lambda + \mathcal{O}(\varepsilon^2). \qquad (1.3)$$

Note that $\mathcal{O}(\varepsilon^2)$ is better than $\mathcal{O}(\varepsilon)$ since we consider small ε.

Rayleigh quotient

Consider the two matrices

$$A_1 = \begin{bmatrix} 2 & 5 \\ 0 & 3 \end{bmatrix} \text{ and } A_2 = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}.$$

Both matrices have an eigenvector $x = \left[\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right]$ with eigenvalue $\lambda = 2$, but A_2 is symmetric. The example code below illustrates that the Rayleigh quotient is much closer to the eigenvalue for the symmetric matrix A_2.

```
>> A1=[2 0;0 3];
>> A2=[2 5;0 3];
>> x=[1;0];
>> y=[1;1]; e=1e-4 % small perturbation
>> z=x+e*y;
>> z'*A1*z/(z'*z)
ans =
   2.000499959998002
>> z'*A2*z/(z'*z)
ans =
   2.000000009998000
```

Theorem 1.1.2 (Accuracy of the Rayleigh quotient). *Suppose (λ, x) is an eigenpair of A with $\|x\| = 1$. Let $v = x + \varepsilon\Delta$ where $\varepsilon \in \mathbb{R}$ and $\|\Delta\| = 1$. Then, for sufficiently small ε*

$$r(v) - \lambda = \begin{cases} \mathcal{O}(\varepsilon^2) & \text{if } x^T A = \lambda x^T \\ \mathcal{O}(\varepsilon) & \text{otherwise.} \end{cases}$$

Note that if (λ, x) is an eigenpair of A and A is symmetric we have that $A^T x - \lambda x = 0$ whose transpose is $x^T A - \lambda x^T = 0$. Therefore, by Theorem 1.1.2 the accuracy is quadratic in ε for symmetric matrices.

Proof. First expand the Rayleigh quotient with the approximation

$$r(v) \quad = \quad \frac{(x+\varepsilon\Delta)^T A(x+\varepsilon\Delta)}{(x+\varepsilon\Delta)^T(x+\varepsilon\Delta)} \quad = \quad \frac{1}{x^Tx+\varepsilon\alpha}(x^TAx \,+\, \varepsilon(\beta \,+\, \varepsilon\gamma))$$

where $\alpha = x^T\Delta + \Delta^T x + \varepsilon\Delta^T\Delta$, $\beta = x^TA\Delta + \Delta^TAx$ and $\gamma = \Delta^TA\Delta$. We will now use the Taylor expansion of functions of the form

$$\frac{1}{1+z} = 1 - z + z^2 - \cdots.$$

By selection $z = \varepsilon\alpha$ and noting that $x^Tx = 1$ by assumption and using that $Ax - \lambda x = 0$, we conclude that

$$r(v) = \lambda + \varepsilon(x^TA - \lambda x^T)\Delta + \mathcal{O}(\varepsilon^2)$$

which reduces to the statement of the theorem. □

1.1.2 Basic eigenvalue methods

The Rayleigh quotient provides a procedure to numerically compute an eigenvalue approximation given an eigenvector approximation. Computing the eigenvector can be done in many ways. We first consider three basic algorithms.

Read about these methods in TB pages 202-209.

- Power method (power iteration) summarized in Algorithm 1
- Inverse iteration summarized in Algorithm 2
- Rayleigh quotient iteration summarized in Algorithm 3
- Rayleigh-Ritz method is summarized in Algorithm 4

1.1.3 Power method

The power method (or sometimes power iteration), is our first eigenvalue method. It consists of starting vector a vector v_0, we multiply this vector with A, scale the resulting vector and repeat the process:

$$v_{k+1} = \alpha_k A v_k, \quad k = 0, \ldots.$$

The scaling factor α_k is used to prevent the iteration values v_k to become very small or very large which makes them more difficult to represent/store. (More precisely, we want to avoid overflow or underflow in the IEEE floating point arithmetic.) Typically the scaling is selected such that $\|v_{k+1}\| = 1$, which can be achieved by setting

$$\alpha_k = \frac{1}{\|Av_k\|}.$$

An important property of the power method is that the only way we need to access the matrix A is in combination with a multiplication with a vector Ax: a so-called *matrix-vector product*. In many scientific applications, the matrix A may be so large that it is not possible to store it explicitly, but the matrix-vector product may still be available.

The operations can be re-ordered such it only requires one matrix vector product per iteration as in Algorithm 1.

If we consider v_k as our eigenvector approximation, we can use the Rayleigh quotient to extract an eigenvalue approximation. Since $\|v_k\|_2^2 = v_k^T v_k = 1$, the Rayleigh quotient reduces to

$$\tilde{\lambda}_k = \frac{v_k^T A v_k}{v_k^T v_k} = v_k^T A v_k.$$

```
Input: A starting vector $v$ with $\|v\| = 1$
Output: Eigenpair approximation $(w, \tilde{\lambda})$
for $n = 1, 2, \ldots$ do
    $w = Av$
    $v = w/\|w\|$
    $\tilde{\lambda} = v^T A v$
end
```

Algorithm 1: Power method (Power iteration).

1.1.4 *Convergence of the power method*

It turns out that the iterates v_k generated by the power method do indeed in general converge to an eigenvector. Under certain (not very restrictive) conditions one can show that

$$\tilde{\lambda}_k = \lambda_1 + \mathcal{O}(\frac{|\lambda_2|^k}{|\lambda_1|^k}).$$

where we have ordered the eigenvalues as $|\lambda_1| \geq |\lambda_2| \geq \cdots$.

Theorem 1.1.3. *Consider a matrix $A \in \mathbb{C}^{n\times n}$, and assume its largest eigenvalue is distinct in modulus such that*

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \cdots \geq |\lambda_n|.$$

If $A = XDX^{-1}$ is a Jordan decomposition of A with $D(1,1) = \lambda$. Suppose the power method is initiated such that the first element of $\mathrm{X}^{-1}v_0$ is non-zero. Then,

$$|\tilde{\lambda}_k - \lambda_1| = \mathcal{O}(\frac{|\lambda_2|^k}{|\lambda_1|^k}).$$

Proof. The power method iterates can be written as

$$v_k = \frac{A^k v_0}{\|A^k v_0\|} \tag{1.4}$$

We can express $a = \mathrm{X}^{-1}v_0$ such that $v_0 = a_1 x_1 + \cdots a_n x_n$. Hence, $A^k v_0 = a_1\lambda_1^k x_1 + \cdots + a_n \lambda_n^k x_n$. We can now factorize $a_1\lambda_1^k$ from the numerator and denominator in (1.4) and obtain

$$v_k = \frac{a_1\lambda_1^k}{|a_1\lambda_1^k|}\frac{x_1 + \varepsilon_k}{\|x_1 + \varepsilon_k\|}$$

where $\varepsilon_k = \frac{a_2\lambda_2^k}{a_1\lambda_1^k}x_2 + \cdots + \frac{a_n\lambda_n^k}{a_1\lambda_1^k}x_n$. The conclusion follows from the fact that $\varepsilon_k \to 0$, when $|\lambda_1| > |\lambda_j|$ for $j = 2,\ldots,n$. □

1.1.5 *Inverse iteration*

Unlike the power method, inverse iteration does not involve a matrix vector product with A per iteration, but one solution to the linear system, $(A-\mu I)^{-1}v$. This operation is normally called a *linear solve*. A linear solve is in general much more computationally expensive than one matrix vector product.

For the next algorithm we form a combination of what we know for the power method, and the observation that the eigenvalues of the matrix A and the matrix

$$B = (A - \mu I)^{-1}$$

are related by a simple relation.

If we denote $\lambda_i(A)$, $\lambda_i(B)$ the eigenvalues of A and B respectively, the eigenvalues are related by

$$\lambda_i(B) = \frac{1}{\lambda_i(A) - \mu} \tag{1.5}$$

The eigenvectors remain unchanged.

The transformation (1.5) has the useful property that the eigenvalues close to μ will be transformed to large eigenvalues. Since inverse iteration converges to the eigenvector corresponding to the largest eigenvalue in general, we obtain with the application of the power method to B, which converges to the eigenvector corresponding to an eigenvalue of A, closest to μ. The eigenvector extraction can be done with the Rayleigh quotient of A, rather than B, as shown in Algorithm 2.

The convergence follows from the fact that the method is equivalent to the power method applied to the matrix B:

$$\tilde{\lambda}_k = \lambda_J + \mathcal{O}\left(\frac{|\lambda_J - \mu|^k}{|\lambda_K - \mu|^k}\right) \tag{1.6}$$

where λ_J is the eigenvalue of A that is closest to μ and λ_K is the eigenvalue of A second closest to μ.

An interpretation of (1.6): The convergence factor of inverse iteration is proportional to the distance between the shift and the closest eigenvalue. In formulas convergence to the eigenvector v is

$$\|v_{k+1} - v\| = \mathcal{O}(|\lambda_J - \mu|\|v_k - v\|)$$

```
Input: A starting vector $v$ with $\|v\| = 1$ and shift $\mu$
Output: Eigenpair approximation $(w, \tilde{\lambda})$
for $n = 1, 2, \ldots$ do
    Solve linear system $(A - \mu I)w = v$
    $v = w/\|w\|$
    $\tilde{\lambda} = v^T A v$
end
```

Algorithm 2: Inverse iteration

1.1.6 *Rayleigh quotient iteration*

For the next basic algorithm, we use a combination of previous ideas. We can set μ in inverse iteration as the Rayleigh quotient. This implies that when the eigenvector is a good approximation, the corresponding eigenvalue of $B = (A - \mu I)^{-1}$ will be a very large value and therefore converge faster than constant μ.

The theoretical convergence of Rayleigh quotient iteration can also be determined by combining results above. In this setting, we will simplify the inverse iteration convergence theory by noting that one step essentially multiplies the previous error with the convergence factor which is $\lambda - \mu$:

$$v_{k+1} - v = \mathcal{O}(\|v_k - v\||\lambda - \mu|). \tag{1.7}$$

Formally, this can be derived from (1.6). Suppose now that we initiate Rayleigh quotient iteration with error ε:

The relationship between (1.6) and (1.7) is consequence of linear convergence. A method which converges linearly with convergence factor α can be described in two equivalent ways

- $v_k - v = \mathcal{O}(\alpha^k)$
- $v_{k+1} - v = \mathcal{O}(\alpha|v_k - v|)$

$$\|v_k - v\| = \varepsilon$$

We compute the eigenvalue approximation with the Rayleigh quotient. The error of the Rayleigh quotient is given by Theorem 1.1.2. Therefore here we have

$$\lambda_k - \lambda = O(\|v_k - v\|^p) = O(\varepsilon^p).$$

Note that the Rayleigh qoutient iteration can also be used for non-symmetric matrices, although it is sometimes presented only as a method for symmetric matrices.

Subsequently, the next eigenvector approximation is computed with inverse iteration whose error is propagated as (1.7):

$$v_{k+1} - v = O(\|v_k - v\||\lambda - \lambda_k|) = O(\varepsilon^{p+1}) = \begin{cases} \mathcal{O}(\varepsilon^3) & \text{if } x^T A = \lambda x^T \\ \mathcal{O}(\varepsilon^2) & \text{otherwise.} \end{cases}$$

In the symmetric case, the error has reduced from ε to ε^3, and the method has cubic convergence for symmetric matrices.

However, in contrast to inverse iteration and the power method, the convergence theory does not determine to which eigenvalue the method converges; it highly depends on the starting vector.

```
Input: A starting vector $v$ with $\|v\| = 1$ and starting eigenvalue $\mu$
Output: Eigenpair approximation $(w, \tilde{\lambda})$
Set $\tilde{\lambda} = \mu$
for $n = 1, 2, \ldots$ do
    Solve linear system $(A - \tilde{\lambda} I)w = v$
    $v = w/\|w\|$
    $\tilde{\lambda} = v^T A v$
end
```

Algorithm 3: Rayleigh Quotient Iteration

1.1.7 *Rayleigh-Ritz method*

In basic linear algebra, we learn that two vectors $x, y \in \mathbb{R}^n$ are orthogonal when $y^T x = 0$. The concept of orthogonality, and its generalization to matrices is very important in this course. We will use it mostly in different factorizations and decompositions of matrices.

The following method is based on a subspace. More precisely, we assume that we have a subspace given, and that we represent it as an orthogonal matrix $Q \in \mathbb{R}^{n\times m}$ where $n > m$. We consider the subspace

$$\mathrm{span}(q_1, \ldots, q_m)$$

where $Q^T Q = I$.

The use of decompositions has been selected as one of the most influential concepts in algorithms in the 20th century: `https://www.siam.org/pdf/news/637.pdf` In this course we also cover other algorithms in the list of important algorithms.

The method can be justified from an idealized reasoning, similar to the Rayleigh quotient. We assume that an eigenvector x lies in the subspace. That is, we assume there exists a vector z such that

$$x = Qz.$$

For the moment we assume that the orthogonal matrix Q is given, and we will see in later sections how we obtain an appropriate subspace and an associated orthogonal matrix.

If we insert this into the eigenvalue equation (1.1) and multiply from the left with Q^T, we obtain

$$Q^T A Q z = \lambda Q^T Q z.$$

Since Q is an orthogonal matrix we see that

$$Q^T A Q z = \lambda z. \tag{1.8}$$

In other words, (z, λ) is an eigenpair of the matrix $H = Q^T A Q$.

The above reasoning was based on *assuming* the eigenvector lies in the subspace. This is obviously not true in general, but it forms as a justification for an algorithm (Algorithm 4) that we expect to return an approximate eigenvalue if the eigenvector almost lies in the subspace. The algorithm contains the computation of the eigenpairs of an $m \times m$-matrix. Assuming that m is much smaller then n, this can be done relatively cheaply, for example with the command `eig` in MATLAB or `eigen` or `eigvals` in Julia.

The Rayleigh-Ritz method is a generalization of the Rayleigh quotient. We obtain the Rayleigh quotient if set $m = 1$.

The computation of eigenvalues of small dense matrices is typically done with the QR-method, the topic of Block 3 in this course. An efficient implementation of the QR-method is available in eig and eigen.

```
Input: Subspace represented by an orthogonal matrix $Q \in \mathbb{R}^{n\times m}$
Output: Eigenpair $m$ approximations $(w, \tilde{\lambda})$
Compute $H = Q^T A Q \in \mathbb{R}^{m\times m}$.
Let $(z, \mu)$ be the $m$ eigenpairs of $H$
Return eigenpair approximations $(w, \tilde{\lambda}) = (Qz, \mu)$
```

Algorithm 4: Rayleigh-Ritz procedure

Remark 1.1.4 (Finite element method viewpoint)**.** *The Rayleigh-Ritz method is in fact the same as the construction with test-space and trial-space in the finite element method. We assume the solution lies in a space and impose a Galerkin condition by enforcing that $Ax - \lambda x$ is orthogonal to Q.*

Proof of Rayleigh-Ritz method

The above reasoning only served as a natural justification of the Rayleigh-Ritz method. A rigorous explanation follows directly from a perturbation theory result in linear algebra, which we provide without a proof.

The Bauer-Fike theorem describes how much the eigenvalues move when we change the matrix by adding ϵB. Most importantly in this setting is that the eigenvalues move with the order of magnitude ϵ.

Theorem 1.1.5 (Bauer-Fike theorem)**.** *Let A be a diagonalizable matrix $A = V\Lambda V^{-1}$ and let μ be an eigenvalue of $A+\epsilon B$. Then,*

$$|\lambda - \mu| \le |\epsilon| \|V\| \|V^{-1}\| \|B\| = \mathcal{O}(\epsilon)$$

for at least one eigenvalue λ of A.

We use the following setting:

- Given an orthogonal Q, let (μ, z) be an eigenpair of $H = Q^TAQ$ from the Rayleigh-Ritz method where $z^Tz = 1$.
- Let (λ, x) be an eigenpair of A, typically with eigenvalue close to μ.

We can now define an error term in the eigenvector approximation ϵy error such that

$$\epsilon y = x - Qz \tag{1.9}$$

The right-hand side of (1.9) is the eigenvector error of the Rayleigh-Ritz method since x is the eigenvector (eigenvector of A) and Qz is the eigenvector approximation from the Rayleigh-Ritz method.

where $\|y\| = 1$. In the following we will show that the error of the Rayleigh-Ritz method is proportional to ϵ, formally justifying why the method works well when an eigenvector almost lies in the subspace Q.

Starting by inserting (1.9) into the eigenvalue equation (1.1) we obtain the following equalities.

$$A(Qz+\epsilon y) = \lambda(Qz+\epsilon y) \tag{1.10}$$

$$AQz + \epsilon(A-\lambda I)y = \lambda Qz \tag{1.11}$$

$$AQz + \epsilon(A-\lambda I)yz^Tz = \lambda Qz \tag{1.12}$$

$$AQz + \epsilon((A-\lambda I)yz^T)z = \lambda Qz \tag{1.13}$$

$$Q^TAQz + \epsilon Q^T((A-\lambda I)yz^T)z = \lambda Q^TQz = \lambda z \tag{1.14}$$

$$(H+\epsilon B)z = \lambda z \tag{1.15}$$

Rearrange the terms

Use that z is normalized: $z^Tz = 1$

Change order of parenthesis

Multiply from the left with Q^T.

In the last step we defined the matrix B from the second term in (1.14). The last equation (1.15) relates the matrix H with the exact eigenvalue λ. Since this equation is exactly of the form in the Bauer-Fike theorem, we conclude that we have an eigenvalue of H near λ, which is at most of distance $\mathcal{O}(\varepsilon)$. That is, if we let μ be an eigenvalue of H, we have

$$\mu = \lambda + \mathcal{O}(\varepsilon).$$

1.2 *Orthogonalization methods*

The Gram-Schmidt procedure is often explained as a procedure to orthogonalize vectors, meaning that given vectors stored in a matrix $F = [f_1, \dots, f_m] \in \mathbb{R}^{n \times m}$ with $n \geq m$ we try to determine $q_1, \dots, q_n$ such that $q_1, \dots, q_n$ are orthonormal and

$$\operatorname{span}(f_1, \dots, f_m) = \operatorname{span}(q_1, \dots, q_m).$$

Such vectors $q_1, \dots, q_n$ exist if $f_1, \dots, f_m$ are linearly independent vectors. Note that the matrix $Q = [q_1, \dots, q_m] \in \mathbb{R}^{n \times m}$ is orthogonal in the sense of definition of orthogonal matrices (see background.pdf).

The Gram-Schmidt procedure can be directly derived by inductively applying the following result.

You have normally learned about the Gram-Schmidt procedure in basic linear algebra courses. We repeat it in a slightly different notation than normal (using orthogonal matrices). It turns out that the classical Gram-Schmidt is not always satisfactory.

In numerical linear algebra, the Gram-Schmidt procedure directly derived from Lemma 1.2.1 is typically called the *classical* Gram-Schmidt procedure in order to distinguish it from variants we discuss later.

Lemma 1.2.1. *Suppose* $Q = [q_1, \dots, q_m] \in \mathbb{R}^{n \times m}$ *is an orthogonal matrix and suppose* $b \notin \operatorname{span}(q_1, \dots, q_m)$. *Let*

$$h = Q^T b$$

and

$$z = b - Qh = (I - QQ^T)b. \tag{1.16}$$

Let $\beta = \|z\|$ *and define*

$$q_{m+1} := \frac{z}{\beta} \tag{1.17}$$

Then,

(a) *the matrix* $[q_1, \dots, q_{m+1}]$ *is an orthogonal matrix;*

(b) $b = h_1 q_1 + \dots + h_m q_m + \beta q_{m+1}$; *and*

(c) $\operatorname{span}(q_1, \dots, q_{m+1}) = \operatorname{span}(q_1, \dots, q_m, b)$.

The vector $h \in \mathbb{R}^n$ is typically referred to as the Gram-Schmidt coefficients.

Proof. Proof of (b): This is a direct consequence of (1.16) and (1.17).
Proof of (a): Note that

$$[q_1, \dots, q_{m+1}]^T [q_1, \dots, q_{m+1}] = [Q, q_{m+1}]^T [Q, q_{m+1}] = \begin{bmatrix} Q^T Q & Q^T q_{m+1} \\ q_{m+1}^T Q & q_{m+1}^T q_{m+1} \end{bmatrix}$$

The conclusion (a) follows from the fact that $Q^T Q = I$,

$$Q^T q_{m+1} = Q^T (I - QQ^T) b = 0$$

and $q_{m+1}^T q_{m+1} = 1$.
Proof of (c): In this course we will several times use the general property that if two rectangular matrices $W \in \mathbb{R}^{n \times m}$ and $V \in \mathbb{R}^{n \times m}$ are related by

$$W = VP \tag{1.18}$$

for some non-singular matrix $P \in \mathbb{R}^{m\times m}$, then then $\operatorname{span}(W) = \operatorname{span}(V)$. If we select P as

$$P = \begin{bmatrix} I & h \\ 0 & \|z\| \end{bmatrix}$$

then (1.18) is satisfied with $V = [Q, q_{m+1}]$ and $W = [Q, b]$. □

Classical Gram-Schmidt example

```
>> Q=(1/sqrt(2))*[1 -1; 1 1; 0 0; 0 0];
>> Q'*Q   % Check if Q is orthogonal
ans =
    1.0000         0
         0    1.0000
>> b=randn(4,1);
>> h=Q'*b;             % Compute Gram-Schmidt coefficients
>> z=b-Q*h;            % Compute "orthogonal complement"
>> beta=norm(z);
>> q_new=z/beta;
>> Q_new=[Q,q_new];    % Construct new Q-matrix
>> Q_new'*Q_new        % Check that Q_new is orthogonal
ans =
    1.0000         0         0
         0    1.0000         0
         0         0    1.0000
>> norm(Q_new*[eye(2), h; zeros(1,2), norm(z)]-[Q,b])
>> P=[eye(2), h; zeros(1,2), beta];
>> norm(Q_new*P-[Q,b])  % Check that span(Q_new)=span([Q,b])
ans =
   1.1444e-16
```

○

Although the above example suggests that classical Gram-Schmidt works, it will in general not be satisfactory in our context. It turns out that the classical Gram-Schmidt is very sensitive to round-off errors in certain situations.

A detailed analysis of the influence of the round-off errors can be found in (appendix) Section 1.7 from which extract one conclusion. If the vector to be orthogonalized is almost in the subspace, we are likely to obtain a large error. Suppose $b = q + \delta e$ where $q \in \operatorname{span}(Q)$ (meaning there $q = Qd$) and $e \perp Q$ and $\|e\| = 1$, for a small δ. Then, the round-off error is

$$\frac{|\varepsilon|}{|\delta|}\|Qd\| + \mathcal{O}(\varepsilon^2).$$

In practice, we have round-off errors in every floating point operation and a complete round-off error analysis is quite cumbersome. In our simplified analysis we assume that no error is intruduced in the computation of z and $\tilde{q}_{m+1}$. In particular, no additional round-off error is introduced in (1.30) and (1.31).

which suggests that the round-off error is proportional to $|\varepsilon|/|\delta|$, and can be very large if δ is very small.

Conclusion error analysis of classical Gram-Schmidt method. The Gram-Schmidt procedure is likely to have a large round-off error if the vector b almost lies in the subspace $\mathrm{span}(Q)$.

Modified Gram-Schmidt

In this course we consider two variations of Gram-Schmidt which aim to improve the floating-point arithmetic problems described above.

We now derive the algorithm called *the modified Gram-Schmidt procedure* from the classical Gram-Schmidt procedure. For theoretical purposes we express the classical Gram-Schmidt in for-loops:

The modified Gram-Schmidt procedure is equivalent to the classical Gram-Schmidt procedure in exact arithmetic, but different floating-point arithmetic.

```
for i=1:m
  h(i)=Q(:,i)’*b;
end
z=b;
for i=1:m
  z=z-h(i)*Q(:,i)
end
```

Note that at iteration i of the second loop, we only need `h(i)` computed at the ith iteration the first loop such that we can merge the two loops:

Although modified Gram-Schmidt yields a different result in floating point arithmetic, it is not always clear that the result is better. In fact, theoretical understanding for this is still disputed by some scientists. You will investigate this in practice by for a specific situation in the homeworks.

```
z=b;
for i=1:m
  h(i)=Q(:,i)’*b;
  z=z-h(i)*Q(:,i);
end
beta=norm(z);
z=z/beta;
```

In the first step inside the for-loop, the vector `z` can be explicitly expressed as:

Caution regarding terminology: In this course we consider $Q \in \mathbb{R}^{n\times m}$ as an orthogonal matrix and want to orthogonalize b which result in algorithms above. In some literature (such as TB) Gram-Schmidt procedures are described for orthogonalizing an entire matrix $A \in \mathbb{R}^{n\times(m+1)}$.

- Iteration $i = 1$: $z = b$
- Iteration $i = 2$: $z = b - h_1 q_1$
- $\vdots$
- Iteration $i = m$: $z = b - h_1 q_1 - \cdots - h_m q_{m-1}$

Now recall that the vectors $q_1, \ldots, q_m$ are assumed to be orthogonal. The following identies can be directly identified.

- Iteration $i = 1$: $q_i^T z = q_i^T b$

- Iteration $i = 2$: $q_i^T z = q_2^T(b - h_1 q_1) = q_2^T b - h_1 q_2^T q_1 = q_i^T b$
- $\vdots$
- Iteration $i = m$: $q_i^T z = q_m^T b - h_1 q_m^T q_1 - \cdots - h_m q_m^T q_{m-1} = q_m^T b = q_i^T b$

Note that for every iteration we have $q_i^T z = q_i^T b$. Therefore, we can replace `Q(:,i)’*b` with `Q(:,i)’*z` in the for-loop. This is what we call the modified Gram-Schmidt method.

```
z=b;
for i=1:m
  h(i)=Q(:,i)’*z;
  z=z-h(i)*Q(:,i)
end
beta=norm(z);
```

Double Gram-Schmidt

The next approach to improve the classical Gram-Schmidt procedure is very naive. Since we know that round-off errors will make the vector $z = b - Qh$ to not be orthogonal in practice, we can try to make it orthogonal by applying classical Gram-Schmidt again. This is what is called repeated Gram-Schmidt, or the special case double Gram-Schmidt.

```
>>  h=Q’*b;
>>  z=b-Q*h;
>>  g=Q’*z;
>>  z=z-Q*g
>>  h=h+g;
>>  beta=norm(z);
>>  z=z/norm(z);
```

1.3 *Krylov methods*

The power method was the basis of both inverse iteration and Rayleigh quotient iteration. These algorithms can be used to compute one eigenvector given an initial guess. In order to compute several eigenvalues we now extend the power method in a different way. We consider the space spanned by the iterates of the power method.

Definition 1.3.1 (Krylov subspace)**.** *The span of the iterates of the power method is called a Krylov subspace*

$$\mathcal{K}_m(A, b) := \operatorname{span}(b, Ab, A^2b, \ldots, A^{m-1}b).$$

Due to rounding error issues, the Krylov subspace is usually not computed from $[b, Ab, A^2b, \dots, A^{m-1}b]$, but rather represented with an orthogonal basis of $\mathcal{K}_m(A, b)$. The Arnoldi method can be seen as method to compute an orthogonal basis of a Krylov subspace. More precisely, the Arnoldi method is a method which generates an orthogonal matrix $Q_m \in \mathbb{C}^{n\times m}$ such that

$$AQ_m = Q_{m+1}\underline{H}_m$$

where $\underline{H}_m \in \mathbb{R}^{(m+1)\times m}$ and $Q_{m+1} = [Q_m, q_{m+1}]$. The matrix $\underline{H}_m$ is a so-called Hessenberg matrix, which means that the elements below the first lower off-diagonal are zero:

$$\underline{H}_m = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

H_m

The Arnoldi method can be used to compute many quantites. In the context of eigenvalue computations, we take the eigenvalues of $H_m \in \mathbb{C}^{m\times m}$, which is the top part of $\underline{H}_m$, as eigenvalue approximations.

1.3.1 Derivation of the Arnoldi method

We will use the Arnoldi factorization as a tool to derive and analyze the Arnoldi method. The Arnoldi factorization is the relation

$$AQ_m = Q_{m+1}\underline{H}_m. \tag{1.19}$$

The Arnoldi method will be seen as method to compute the Arnoldi factorization, by expanding Q_m and $\underline{H}_m$ to form Q_{m+1} and $\underline{H}_m$. The algorithm can be derived by induction. Suppose and Arnoldi factorization for $m = 2$ is given

$$AQ_2 = Q_3\underline{H}_2 \tag{1.20}$$

and we wish to expand the matrices such that they satisfy

$$AQ_3 = Q_4\underline{H}_3. \tag{1.21}$$

This is a matrix equality and if we consider column $1, 2$ of this equality we obtain exactly (1.20). Column 3 is given by multiplication with e_3:

$$AQ_3e_3 = Q_4\underline{H}_3e_3.$$

We simplify this equation to

$$Aq_3 = q_1 h_{1,3} + q_2 h_{2,3} + q_3 h_{3,3} + q_4 h_{4,3}. \tag{1.22}$$

Note that q_1, q_2, q_3 are known since they form Q_3. It remains to determine $h_{1,3}, \ldots, h_{4,3}$ and q_4. If we denote the left-hand side of (1.22) by b we see that the problem to determine the coefficients is exactly the problem we solved with Gram-Schmidt in Lemma 1.2.1. Therefore, the Arnoldi method essentially consists of applying matrix vector products and carrying out a Gram-Schmidt procedure.

1.3.2 The Arnoldi method

If we combine the (Gram-Schmidt) orthogonalization process with the Krylov subspace, we obtain the algorithm called the Arnoldi method.

```
Input: A starting vector $b$
Output: Eigenpair approximations
Set $q_1 = b/\|b\|$, $H_0$ =empty matrix
for $m = 1, 2, \ldots$ do
    Compute $x = Aq_m$
    Orthogonalize $x$ against $q_1, \ldots, q_m$ by computing $h \in \mathbb{C}^m$ and
      $x_\perp \in \mathbb{C}^n$ such that $Q^T x_\perp = 0$ and
                  $x_\perp = x - Qh.$
    Let $\beta = \|x_\perp\|$
    Let $q_{m+1} = x_\perp/\beta$
    Expand $\underline{H}_{m-1}$ with one column:
                  $\underline{H}_m := \begin{bmatrix} \underline{H}_{m-1} & h \\ 0 & \beta \end{bmatrix}$
end
Compute eigenpairs of $H_m$: $\lambda, z$ (called Ritz pairs)
Return eigenpair approximations $(\lambda, Qz)$.
```

Algorithm 5: Arnoldi's method for eigenvalue problems.

Arnoldi's method for eigenvalue problems is also discussed in TB pages 251–264.

1.3.3 *The Lanczos method*

There are various ways to improve and specialize the Arnoldi method for specific matrix structures. The most prominent specialization is for symmetric matrices and called the Lanczos method. First observe that H_m can be expressed as

$$H_m = Q_m^T A Q_m.$$

By transposing the left-hand side and the right-hand side we obtain

$$H_m^T = (Q_m^T A Q_m)^T = Q_m^T A^T Q_m = Q_m^T A = Q_m = H_m.$$

Hence, H_m is also symmetric. Since H_m is both symmetric and has the Hessenberg structure, it is a tridiagonal matrix. This forms the foundation of the derivation of the Lanczos method.

History: The Arnoldi method was invented by Walter Edwin Arnoldi in 1951 and the Lanczos method is named after the work of Cornelius Lanczos in 1950. In those days, most eigenvalue problems arose in acoustics and vibrations that lead to symmetric matrices, which is one reason the symmetric specialization was invented first. The Krylov method is named after Aleksey Krylov (or Алексе́й Крыло́в) who presented some ideas for Krylov subspaces in the context of naval engineering in 1931. This method class was also deemed the one of the most important algorithms in the 20th century by SIAM - Society of industrical and applied mathematics.

Current research: Large parts of the current international numerical linear algebra research community focus on Krylov methods, with challanges ranging from modern hardware implementations to generalizations for structured eigenproblems in new emerging fields. Search the web for “book of abstracts” and “numerical linear algebra” and “Krylov”.

```
Output: Eigenpair approximations
Input: The matrix $A$ and vector $b$.
$b$ =arbitrary, $q_1 = b/\|b\|$, $H_0$ =empty matrix
for $m = 1,2,\ldots$ do
    $v = Aq_m$
    $\alpha_m = q_m^T v$
    $v = v - \beta_{m-1}q_{m-1} - \alpha_m q_m$
    $\beta_m = \|v\|$
    $q_{m+1} = v/\beta_m$
end
Construct the matrix
```

$$H = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{m-1} \\ & & \beta_{m-1} & \alpha_m \end{bmatrix}$$

Algorithm 6: The Lanczos method

1.4 *Convergence of Arnoldi's method for eigenvalue problems*

Recall that, unless it breaks down, k steps of the Arnoldi method generates an orthogonal basis of a Krylov subspace, represented by a matrix $Q = [q_1, \ldots, q_k] \in \mathbb{C}^{n\times k}$ such that $Q^*Q = I$ and

$$\operatorname{span}(q_1, \ldots, q_k) = \mathcal{K}_k(A, b) := \operatorname{span}(b, Ab, \ldots, A^{k-1}b).$$

The eigenvalue approximations (called Ritz values) are subsequently found from the eigenvalues of

$$H = Q^*AQ.$$

The matrix $H \in \mathbb{C}^{k\times k}$ is a Hessenberg matrix and can be generated as a by-product of the Arnoldi method. We call a pair (μ, Qv) a Ritz pair and Qv a Ritz vector, if v and μ safisfy

$$Hv = \mu v.$$

1.4.1 *Bound for subspace-eigenvector angle*

As a first indicator of the convergence we will characterize the following quantity

$$\text{error in eigenvector } x_i \sim \|(I - QQ^*)x_i\| \tag{1.23}$$

The Lanczos iteration is also described in TB pages 276-278.

where

$$Ax_i = \lambda_i x_i.$$

It is very natural to associate the accuracy of the eigenvector with this quantity from a geometric perspective. The indicator in the right-hand side of (1.23) is called (the norm of) the orthogonal complement of the projection of x_i onto the space spanned by Q and it can be interpreted as the sine of the canonical angle between the Krylov subspace and an eigenvector. For the moment, we will only justify this indicator with this geometric reasoning and the following observation:

Lemma 1.4.1. *Suppose (λ_i, x_i) is an eigenpair A. If the Krylov subspace contains the eigenvector ($x_i \in \mathcal{K}_k(A,b)$), then the indicator vanishes$\|(I - QQ^*)x_i\| = 0$ and there is at least one Ritz value μ such that $\mu = \lambda_i$.*

In words:

Recall: $Q \in \mathbb{C}^{n\times k}$ is an orthogonal matrix which means that $Q^*Q = I \in \mathbb{C}^{k\times k}$. However, $I \neq QQ^* \in \mathbb{C}^{n\times n}$.

- Suppose the Krylov subspace contains the eigenvector ($x_i \in \mathcal{K}_k(A,b)$). Then, there exists a vector $z \in \mathbb{C}^k$ such that $x_i = Qz$. Moreover, this is an eigenvector of H such that the Arnoldi method will generate an exact eigenvalue of A. Moreover, the indicator is $\|(I - QQ^*)x_i\| = \|(I - QQ^*)Qz\| = 0$.
- If, similar to above, $x_i \approx x \in \mathcal{K}_k(A,b)$, we expect the indicator to be small and an eigenvalue of H also to be close λ_i.

The indicator can be bounded as follows, where we assume diagonalizability of the matrix.

Theorem 1.4.2. *Suppose $A \in \mathbb{C}^{n\times n}$ is diagonalizable and let the matrix $X = (x_1, \ldots, x_n) \in \mathbb{C}^{n\times n}$ and diagonal matrix $\Lambda \in \mathbb{C}^{n\times n}$ be the Jordan decomposition such that*

$$A = X\Lambda X^{-1}.$$

Suppose $\alpha_1, \ldots, \alpha_n \in \mathbb{C}\backslash\{0\}$ are such that

$$b = \alpha_1 x_1 + \cdots + \alpha_n x_n \tag{1.24}$$

The Arnoldi method produces an exact approximation if the Krylov subspace contains an eigenvector, or equivalently the indicator is zero.

and

$$\varepsilon_i^{(m)} := \min_{\substack{p \in P_{m-1}\\ p(\lambda_i)=1}} \max(|p(\lambda_1)|, \ldots, |p(\lambda_{i-1})|, |p(\lambda_{i+1})|, \ldots, |p(\lambda_n)|)$$

where P_n denotes polynomials of degree n. Suppose the Arnoldi method does not break down when applied to A and started with b. Let $Q \in \mathbb{C}^{n\times m}$ be the orthogonal basis generated after m iterations. Then,

$$\|(I - QQ^*)x_i\| \le \xi_i \varepsilon_i^{(m)}, \qquad (1.25)$$

Recall: The eigenvectors of a diagonalizable matrix form a basis of $\mathbb{C}^n$.

where

$$\xi_i = \sum_{\substack{j=1\\ j\neq i}}^{n} \frac{|\alpha_j|}{|\alpha_i|}.$$

Proof. The proof consists of three steps.

1. Consider any vector $u \in \mathbb{C}^n$. Then

$$\min_{z\in\mathbb{C}^m} \|u - Qz\|_2$$

is a linear least squares problem with a solution given by the normal equations $Q^*u = Q^*Qz$. Hence, $z = Q^*u$. This implies that (for any vector u) we have

$$\min_{z\in\mathbb{C}^m} \|u - Qz\|_2 = \|u - QQ^*u\| = \|(I - QQ^*)u\|$$

2. Although we ultimately want to bound the left-hand side of (1.25), the proof is simplified by considerations of a scaling the left-hand side of (1.25) with α_i as follows:

$$\begin{aligned}\|(I - QQ^*)\alpha_i x_i\| &= \min_{z\in\mathbb{C}^m} \|\alpha_i x_i - Qz\| \\ &= \min_{y\in\mathcal{K}_m(A,b)} \|\alpha_i x_i - y\|\end{aligned}$$

The indicator can be bounded by a product consisting of two scalar values: $\varepsilon_i^{(m)}$ which only depends on the eigenvalues and iteration number; and ξ_i only depending on the starting vector and eigenvectors.

Apply step 1 reversely with $u = \alpha_i x_i$

Now note that the space $\mathcal{K}_m(A,b)$ can be characterized with polynomials. It is easy to verify that $y \in \mathcal{K}_m(A,b)$ is equivalent to the existance of a polynomial $p \in P_{m-1}$ such that $y = p(A)b$. Consequently,

$$\|(I - QQ^*)\alpha_i x_i\| = \min_{p\in P_{m-1}} \|\alpha_i x_i - p(A)b\|.$$

3. The final step consists of inserting the expansion of b in terms of eigenvectors (1.24) and applying appropriate bounds:

$$
\begin{aligned}
\|(I-QQ^*)\alpha_i x_i\| &= \min_{p\in P_{m-1}} \|\alpha_i x_i - p(A)\sum_{j=1}^n \alpha_j x_j\| \\
&= \min_{p\in P_{m-1}} \left\|\alpha_i x_i - \sum_{j=1}^n \alpha_j p(\lambda_j) x_j\right\| \\
&\le \min_{\substack{p\in P_{m-1}\\ p(\lambda_i)=1}} \left\|\alpha_i x_i - \sum_{j=1}^n \alpha_j p(\lambda_j) x_j\right\| \\
&= \min_{\substack{p\in P_{m-1}\\ p(\lambda_i)=1}} \|\alpha_i x_i - \alpha_i x_i - \sum_{\substack{j=1\\ j\ne i}}^n \alpha_j p(\lambda_j) x_j\| \\
&= \min_{\substack{p\in P_{m-1}\\ p(\lambda_i)=1}} \|\sum_{\substack{j=1\\ j\ne i}}^n \alpha_j p(\lambda_j) x_j\| \\
&\le (\sum_{\substack{j=1\\ j\ne i}}^n |\alpha_j|)\cdot \min_{\substack{p\in P_{m-1}\\ p(\lambda_i)=1}} \max_{j\ne i}(|p(\lambda_j)|) \\
&= (\sum_{\substack{j=1\\ j\ne i}}^n |\alpha_j|)\cdot \varepsilon_i^{(m)}
\end{aligned}
$$

The conclusion of the theorem is established by dividing the equation by $|\alpha_i|$.

□

Note that $\|b\| = 1$ and $\|x_1\| = \cdots = \|x_n\| = 1$. Hence the coefficients $\alpha_1,\ldots,\alpha_n$ are balanced. In particular they satisfy

$$1 = \|\alpha_1 x_1 + \cdots + \alpha_n x_n\| \le |\alpha_1| + \cdots + |\alpha_n|.$$

and

$$\xi_i = \frac{1}{|\alpha_i|}\sum_{j=1}^n |\alpha_j| - 1 \ge \frac{1}{|\alpha_i|} - 1$$

From this we can easily identify a very good situation and a very bad situation.

- Suppose for all $j \ne i$, $\alpha_j = \delta$ and suppose δ is small. We have that $\xi_i = \frac{(n-1)\delta}{\alpha_i}$. Due to balancing α_i cannot be small. Hence, ξ_i is small, showing fast convergence for this eigenvalue.
- On the other hand, if α_i (the component of the starting vector in the direction of the ith eigenvector) is very small, we have $\xi_i \gg 1$ which implies that the right-hand side of (1.25) is large and we have slow convergence.

This serves as a justification for a more general property.

Rule-of-thumb. Starting vector dependency. The Arnoldi method for eigenvalue problems will "favor" eigenvectors which have large components in the starting vector.

The word "favors" is purposely vague. It should be interpreted as the situation that one observes often in practice, but certainly not always. If we have a particular structure in the matrix or starting vector, we might observe convergence to other eigenvalues.

Bounding $\varepsilon_i^{(m)}$

In the characterization of the indicator in Theorem 1.4.2 above we introduced the quantity $\varepsilon_i^{(m)}$. This quantity bounds (up to a constant) the error in eigenvector x_i at iteration m. Although $\varepsilon_i^{(m)}$ is defined through a polynomial optimization problem, which is complicated to solve, it is surprisingly easy to use this to obtain bounds providing qualitative understanding of the convergence of the Arnoldi method for eigenvalue problems. We illustrate the power with a specific bound.

Since x_i eigenvector, $p(A)x_i = p(\lambda_i)x_i$

Corollary 1.4.3. *Suppose $C(\rho, c) \subset \mathbb{C}$ is a disk centered at $c \in \mathbb{C}$ with radius ρ such that it contains all eigenvalues but λ_1. That is, $\lambda_2, \ldots, \lambda_n \in C(\rho, c)$ and $\lambda_1 \notin C(\rho, c)$. Then,*

$$\varepsilon_1^{(m)} \le \left(\frac{\rho}{|\lambda_1 - c|}\right)^{m-1}.$$

For any two sets $S \subset Z$: $\min_{z\in Z} g(z) \le \min_{z\in S} g(z)$

Think: $\varepsilon_i^{(m)}$ measures how "difficult" it is to push down a polynomial in points λ_j, for all $j \neq i$ and maintain $p(\lambda_i) = 1$.

Proof. The proof consists of selecting a particular polynomial in the polynomial optimization problem,

$$\begin{aligned}\varepsilon_1^{(m)} &:= \min_{\substack{p\in P_{m-1}\\ p(\lambda_1)=1}} \max(|p(\lambda_1)|, \ldots, |p(\lambda_{i-1})|, |p(\lambda_{i+1})|, \ldots, |p(\lambda_n)|)\\ &= \max_{j\neq i} |q(\lambda_j)|,\end{aligned}$$

for any $q \in P_{m-1}$ satisfying $q(\lambda_1) = 1$, in particular

$$q(z) = \frac{1}{(\lambda_1 - c)^{m-1}}(z - c)^{m-1}.$$

Hence, from the definition of ρ and c we have that

$$\begin{aligned}\varepsilon_1^{(m)} &\le \max_{i>1} \frac{|\lambda_i - c|^{m-1}}{|\lambda_1 - c|^{m-1}}\\ &\le \frac{\rho^{m-1}}{|\lambda_1 - c|^{m-1}}.\end{aligned}$$

□

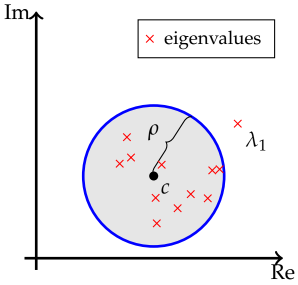
The result can be intuitively interpreted as follows. If we can construct a small disc that encloses all eigenvalues but one eigenvalue we expect fast (at least linear geometric) convergence for that eigenvalue. This can be achieved for an eigenvalue which is well separated from the rest of the eigenvalues and also in an outer part of the spectrum. We call this "extreme" isolated eigenvalues.

Rule-of-thumb. Eigenvalue dependency. Arnoldi's method for eigenvalue problems favors convergence to "extreme" isolated eigenvalues.

Note the difference between an "extreme" eigenvalue and the eigenvalues which are largest in modulus (absolute value). The Arnoldi method will favor "extreme" whereas the power method will essentially always converge to the eigenvalue largest in modulus.

1.4.2 *An a posteriori theorem*

In the previous section we saw a characterization of the error involving the eigenvectors and eigenvalues of the matrix A. The following result provides an explicit characterization of $\|Av-\mu v\|$ where (μ,v) is an approximate eigenpair. It is expressed in terms of quantities computed during the iteration.



Theorem 1.4.4. *Suppose Q_k and $\underline{H}_k$ satisfy the Arnoldi relation*

$$AQ_k = Q_{k+1}\underline{H}_k \tag{1.26}$$

where $Q_k \in \mathbb{C}^{n\times k}$ and $Q_{k+1} = [Q_k, q_{k+1}] \in \mathbb{C}^{n\times(k+1)}$ are orthogonal matrices. Moreover, suppose (μ, v) is a Ritz pair such that $H_k z = \mu z$ and $v = Q_k z$. Then,

$$\|Av-\mu v\|_2 = |h_{k+1,k}||e_k^T z|. \tag{1.27}$$

Proof. From the fact that (μ,v) is a Ritz pair, we have

$$\begin{aligned} Av-\mu v &= AQ_k z - \mu Q_k z \\ &= (AQ_k - Q_k H_k)z \\ &= h_{k+1,k} q_{k+1} e_k^T z \end{aligned}$$

The conclusion follows from the fact that $e_k^T z$ is a scalar and q_{k+1} is normalized since Q_{k+1} is orthogonal. More precisely, $\|Av-\mu v\|_2 = |h_{k+1,k}|\|q_{k+1}e_k^T z\| = |h_{k+1,k}|\|q_{k+1}\||e_k^T z| = |h_{k+1,k}||e_k^T z|$. □

A priori vs. a posteriori: Error characterizations can be classified into two types. An *a priori* (latin for "from before") error estimate involves quantities which are known before the algorithm is carried out. An *a posteriori* (latin for "from after") error characterization involves quantites computed during the iteration. Theorem 1.4.2 is an a priori error bound. Theorem 1.4.4 is an (exact) a posteriori error characterization since the right-hand side involves H_k and z which are computed from the iteration.

The result can be used to study break-down. Break-down corresponds to the situation where we cannot carry out that Gram-Schmidt orthogonalization process since the new vector is contained in the span of previous iterations. It implies that the $y_\perp = 0$ and $\beta = 0$. This implies in turn that $h_{k+1,k} = 0$. Hence, due to (1.27), if we have breakdown the error is already zero and the Ritz pairs are eigenpairs of the original problem.

1.5 *Shift-and-invert Arnoldi method*

We saw that the convergence of the Arnoldi method was given in terms of a polynomial optimization, which in turn gave bounds on the convergence factor; from which we conclude favorable convergence for the outer part of the spectrum. In an application, this may not necessarily be the eigenvalues of interest. This sitution is similar to the power method. Similar to the construction of inverse iteration we can transform the problem and use the Arnoldi method on a matrix:

$$B = (A - \mu I)^{-1},$$

where μ is called a shift (or target). This is called the shift-and-invert Arnoldi method.

Properties:

- The shift-and-invert Arnoldi method requires a linear solve per iteration, in contrast to the standard Arnoldi method which requires a matrix-vector multiplication.

- The convergence if shift-and-invert Arnoldi method is completely given by the convergence of the Arnoldi method with the transformed matrix B.

- In contrast to inverse iteration, which is essentially guaranteed to converge to the closest eigenvalue, the shift-and-invert Arnoldi method in practice often converges to eigenvalues close to μ, but the precise relationship is more complicated, since the convergence of the Arnoldi method is more complicated than the convergence of the power method.

- The eigenvalue extraction in shift-and-invert Arnoldi method can be done in different ways. The standard approach to extract eigenvalue approximations is to use H_m such that

  $$(A - \mu I)^{-1} Q_m = Q_{m+1} \underline{H}_m. \tag{1.28}$$

  Another approach is to use $G_m = Q_m^T A Q_m$, where Q_m is generated from the Arnoldi method applied to $(A - \mu I)$.

1.6 *Literature and further reading*

The proof and reasoning above is inspired by [5]. Other convergence bounds involving Schur factorizations, that lead to similar qualitative understanding can be found in [6], where also complications of the non-generic cases are discussed. There are also further characterizations of convergence and the connection with potential theory [4]. In the above reasoning we characterized the angle between the subspace and the eigenvector. Although this serves as a very accurate prediction of the error in practice, it does not directly give a rigorous bound on the accuracy of Ritz pair. Several approaches to describe the convergence of Ritz values and Ritz vectors have been done in for instance [2, 3]. There is also considerable research on the effect of rounding errors in Krylov methods. Unlike many other numerical methods, the effect of finite arithmetic can improve the performance of the algorithm. See also the recent summary of the convergence of the Arnoldi method for eigenvalue problems [1]. The a posteriori error estimate in Theorem 1.4.4 is contained in some recent text-books in numerical linear algebra such as [7].

1.7 *Appendix: Round-off error analysis of Gram-Schmidt*

We now investigate what happens if we have an error in the computation of the Gram-Schmidt coefficients. In other words, we assume that h is approximated by

$$\tilde{h} = \begin{bmatrix} (1+\varepsilon_1)h_1 \\ \vdots \\ (1+\varepsilon_m)h_m \end{bmatrix} = \left(\begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix} + \underbrace{\begin{bmatrix} \varepsilon_1 & & \\ & \ddots & \\ & & \varepsilon_m \end{bmatrix}}_{\Lambda_\varepsilon}\right) Q^T b \tag{1.29}$$

where $\varepsilon_1,\ldots,\varepsilon_m$ are a small number introduced by the inexact evaluation of $Q^T b$, typically of order of the same order of magnitude ϵ_{mach}. Our approximation of z satisfies

$$\tilde{z} = b - Q\tilde{h} = b - Q\Lambda_\varepsilon Q^T b(1+\varepsilon) = z - Q\Lambda_\varepsilon Q^T b \tag{1.30}$$

such that

$$\tilde{q}_{m+1} = \frac{1}{\|\tilde{z}\|}\tilde{z} = \frac{1}{\|z - Q\Lambda_\varepsilon Q^T b\|}\tilde{z} = \frac{1}{\sqrt{(z - Q\Lambda_\varepsilon Q^T b)^T (z - Q\Lambda_\varepsilon Q^T b)}}\tilde{z} =$$
$$\frac{1}{\sqrt{(z - Q\Lambda_\varepsilon Q^T b)^T (z - Q\Lambda_\varepsilon Q^T b)}}\tilde{z} = \frac{1}{\sqrt{\|z\|^2 + \|\Lambda_\varepsilon\|^2 \|QQ^T b\|^2}}\tilde{z} =$$
$$\tilde{z}\left(\tfrac{1}{\|z\|} + \mathcal{O}(\varepsilon^2)\right), \tag{1.31}$$

where $\varepsilon = \|\Lambda_\varepsilon\|$. The approximation of the new vector is

$$\tilde{q}_{m+1} = (z - Q\Lambda_\varepsilon Q^T b)(\frac{1}{\|z\|} + \mathcal{O}(\varepsilon^2)) = \frac{z}{\|z\|} - \frac{1}{\|z\|}Q\Lambda_\varepsilon Q^T b + \mathcal{O}(\varepsilon^2) \quad (1.32)$$

In this first-order estimation, we see that the error is small if

$$\frac{\|Q\Lambda_\varepsilon Q^T b\|}{\|z\|} = \frac{\|\Lambda_\varepsilon Q^T b\|}{\|z\|} \leq \varepsilon \frac{\|Q^T b\|}{\|z\|}$$

is small.

A bad situation can easily be identified, since we can construct a situation where $\|z\|$ is small but $Q^T b$ is not: Suppose $b = q + \delta e$ where $q = Qd$ and $e \perp Q$ and $\|e\| = 1$. A direct computation leads to

$$\|\tilde{q}_{m+1} - \frac{z}{\|z\|}\| \leq \frac{|\varepsilon|}{|\delta|}\|Qd\| + \mathcal{O}(\varepsilon^2).$$

which suggests that the round-off error is proportional to $|\varepsilon|/|\delta|$.

1.8 Appendix: Round-off error analysis in inverse iteration

The condition number of a matrix, defined by $K(A) := \|A\|\|A^{-1}\|$, gives a description of how much error we can expect when computing B^{-1}. More precisely, using (any numerical) method to compute B^{-1}, we can always expect an error at least of magnitue $K(B)$, due to rounding errors.

Inverse iteration and Rayleigh quotient iteration both involve the matrix $(A - \lambda I)^{-1}$, where fast convergence can be expected if λ is almost an eigenvalue. On the other hand, if λ is almost an eigenvalue, $B := A - \lambda I$ is almost singular, and the condition number $K(B) = K(A - \lambda I)$ is large.

Normally, one would conclude from this reasoning that λ should not be too close to the eigenvalue. However, it turns out that the situation is much better than one can expect, if we assume that we use a backward stable solver to solve to compute b as the numerical solution to the linear system $B^{-1}z \approx x$. Backward stable solver means that the error can be moved into the input, in this case as a modification in the B-matrix. That is, the method to solve the linear system is of such character that there exists a matrix E such that

$$B^{-1}z \approx x = (B + \epsilon E)^{-1}b$$

where ϵ is typically of the same order of magnitude of machine precision. Since inverse iteration involves a computing x and then normalize, we obtain

$$\frac{x}{\|x\|} = \frac{1}{\|(B + \epsilon E)^{-1}b\|}(B + \epsilon E)^{-1}b \quad (1.33)$$

Many methods for linear systems are indeed backward stable in general, for example backslash in matlab and Julia which are based on Gaussian elimination.

When $B = A - \lambda I$ and λ is almost an eigenvalue, the matrix B will indeed be badly conditioned and $\|(B + \epsilon E)^{-1}\| \to \infty$ as (the rounding error) $\epsilon \to 0$. However, in general the error is dominating in the direction of the eigenvector, making this rounding error to not influence the solution substantially.

Let us be more precise. We now use the property that if C is singular matrix with rank $n - 1$, we have that the following expansion is asymptotically a rank-one matrix:

$$\frac{(C + \epsilon E)^{-1}}{f(\epsilon)} = vu^T + O(\epsilon) \tag{1.34}$$

for some scalar smooth function f with $f(0) \neq 0$. The vector v is the right singular vector $Cv =$ and u is the left singular $u^T C = 0$. Essentially, this means that the error is in the direction of the eigenvector.

Due to the normalization, this type of error is benign. If λ_* is an exact eigenvalue and $C = A - \lambda_*$

$$\frac{x}{\|x\|} = v + O(\epsilon + |\lambda - \lambda_*|)$$

assuming $u^T b \neq 0$. For small rounding errors and accurate eigenvalue approximation, we will obtain an accurate eigenvector.

Note: The property (1.34) can be derived from adjugates of matrices. We have $X^{-1} = \operatorname{adj}(X)/\det(X)$. This means that $\det(X)I = X\operatorname{adj}(X)$. This formula also holds if X is singular, and implies that $\operatorname{adj}(X) = \gamma vu^T$ where v is the singular vector of X.

1.9 *References*

[1] M. Bellalij, Y. Saad, and H. Sadok. Further analysis of the Arnoldi process for eigenvalue problems. *SIAM J. Numer. Anal.*, 48(2):393–407, 2010.

[2] Z. Jia. The convergence of generalized Lanczos methods for large unsymmetric eigenproblems. *SIAM J. Matrix Anal. Appl.*, 16(3):843–862, 1995.

[3] Z. Jia and G. W. Stewart. On the convergence of ritz values, ritz vectors, and refined ritz vectors. Technical report, 1999.

[4] A. B. Kuijlaars. Convergence analysis of Krylov subspace iterations with methods from potential theory. *SIAM Rev.*, 48(1):3–40, 2006.

[5] Y. Saad. *Numerical methods for large eigenvalue problems*. SIAM, 2011.

[6] G.W. Stewart. *Matrix Algorithms volume 2: eigensystems*. SIAM publications, 2001.

[7] D. S. Watkins. *Fundamentals of matrix computations. 3rd ed*. Wiley, 2010.