Chapter 12

The Power Method and Related Methods

You may want to visit Linear Algebra: Foundations to Frontiers - Notes to LAFF With [27] (Chapter 12) in which the Power Method and Inverse Power Methods are discussed at a more rudimentary level.

12.1 The Power Method

The Power Method is a simple method that under mild conditions yields a vector corresponding to the eigenvalue that is largest in magnitude.

Throughout this section we will assume that a given matrix $A \in \mathbb{C}^{m\times m}$ is *nondeficient*: there exists a nonsingular matrix X and diagonal matrix Λ such that $A = X\Lambda X^{-1}$. (Sometimes this is called a diagonalizable matrix since there exists a matrix X so that

$$X^{-1}AX = \Lambda \text{ or, equivalently, } A = X\Lambda X^{-1}.$$

From "Notes on Eigenvalues and Eigenvectors" we know then that the columns of X equal eigenvectors of A and the elements on the diagonal of Λ equal the eigenvalues::

$$X = \left(\begin{array}{c|c|c|c} x_0 & x_0 & \cdots & x_{m-1} \end{array} \right) \quad \text{and} \quad \Lambda = \left(\begin{array}{c|c|c|c} \lambda_0 & 0 & \cdots & 0 \\ \hline 0 & \lambda_1 & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & \lambda_{m-1} \end{array} \right)$$

so that

$$Ax_i = \lambda_i x_i \quad \text{for } i = 0, \ldots, m-1.$$

For most of this section we will assume that

$$|\lambda_0| > |\lambda_1| \geq \cdots \geq |\lambda_{m-1}|.$$

In particular, λ_0 is the eigenvalue with maximal absolute value.

12.1.1 First attempt

Now, let $v^{(0)} \in \mathbb{C}^{m\times m}$ be an "initial guess". Our (first attempt at the) Power Method iterates as follows:

$$\begin{array}{l} \text{for } k = 0, \ldots \\ \quad v^{(k+1)} = Av^{(k)} \\ \text{endfor} \end{array}$$

Clearly $v^{(k)} = A^k v^{(0)}$. Let

$$v^{(0)} = Xy = \psi_0 x_0 + \psi_1 x_1 + \cdots + \psi_{m-1} x_{m-1}.$$

What does this mean? We view the columns of X as forming a basis for $\mathbb{C}^m$ and then the elements in vector $y = X^{-1} v^{(0)}$ equal the coefficients for describing $v^{(0)}$ in that basis. Then

$$\begin{array}{rcl} v^{(1)} = Av^{(0)} & = & A\left(\psi_0 x_0 + \psi_1 x_1 + \cdots + \psi_{m-1} x_{m-1}\right) \\ & = & \psi_0 \lambda_0 x_0 + \psi_1 \lambda_0 x_1 + \cdots + \psi_{m-1} \lambda_{m-1} x_{m-1}, \\ v^{(2)} = Av^{(1)} & = & \psi_0 \lambda_0^2 x_0 + \psi_1 \lambda_1^2 x_1 + \cdots + \psi_{m-1} \lambda_{m-1}^2 x_{m-1}, \\ & \vdots & \\ v^{(k)} = Av^{(k-1)} & = & \psi_0 \lambda_0^k x_0 + \psi_1 \lambda_1^k x_1 + \cdots + \psi_{m-1} \lambda_{m-1}^k x_{m-1}. \end{array}$$

Now, as long as $\psi_0 \neq 0$ clearly $\psi_0 \lambda_0^k x_0$ will eventually dominate which means that $v^{(k)}$ will start pointing in the direction of x_0. In other words, it will start pointing in the direction of an eigenvector corresponding to λ_0. The problem is that it will become infinitely long if $|\lambda_0| > 1$ or infinitessimily short if $|\lambda_0| < 1$. All is good if $|\lambda_0| = 1$.

12.1.2 Second attempt

Again, let $v^{(0)} \in \mathbb{C}^{m\times m}$ be an "initial guess". The second attempt at the Power Method iterates as follows:

$$\begin{array}{l} \text{for } k = 0, \ldots \\ \quad v^{(k+1)} = Av^{(k)}/\lambda_0 \\ \text{endfor} \end{array}$$

It is not hard to see that then

$$\begin{array}{rcl} v^{(k)} & = & Av^{(k-1)}/\lambda_0 = A^k v^{(0)}/\lambda_0^k \\ & = & \psi_0 \left(\frac{\lambda_0}{\lambda_0}\right)^k x_0 + \psi_1 \left(\frac{\lambda_1}{\lambda_0}\right)^k x_1 + \cdots + \psi_{m-1} \left(\frac{\lambda_{m-1}}{\lambda_0}\right)^k x_{m-1} \\ & = & \psi_0 x_0 + \psi_1 \left(\frac{\lambda_1}{\lambda_0}\right)^k x_1 + \cdots + \psi_{m-1} \left(\frac{\lambda_{m-1}}{\lambda_0}\right)^k x_{m-1}. \end{array}$$

Clearly $\lim_{k\to\infty} v^{(0)} = \psi_0 x_0$, as long as $\psi_0 \neq 0$, since $\left|\frac{\lambda_k}{\lambda_0}\right| < 1$ for $k > 0$.

Another way of stating this is to notice that

$$A^k = \underbrace{(AA\cdots A)}_{k \text{ times}} = \underbrace{(X\Lambda X^{-1})(X\Lambda X^{-1})\cdots(X\Lambda X^{-1})}_{\Lambda^k} = X\Lambda^k X^{-1}.$$

so that

$$\begin{array}{rcl} v^{(k)} & = & A^k v^{(0)}/\lambda_0^k \\ & = & A^k X y/\lambda_0^k \\ & = & X\Lambda^k X^{-1} X y/\lambda_0^k \\ & = & X\Lambda^k y/\lambda_0^k \\ & = & X\left(\Lambda^k/\lambda_0^k\right) y = X \left(\begin{array}{c|c|c|c} 1 & 0 & \cdots & 0 \\ \hline 0 & \left(\frac{\lambda_1}{\lambda_0}\right)^k & \cdots & 0 \\ \hline \vdots & \ddots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & \left(\frac{\lambda_{m-1}}{\lambda_0}\right)^k \end{array}\right) y. \end{array}$$

Now, since $\left|\frac{\lambda_k}{\lambda_0}\right| < 1$ for $k > 1$ we can argue that

$$\begin{array}{rcl} \lim_{k\to\infty} v^{(k)} & = & \lim_{k\to\infty} X \left(\begin{array}{c|c|c|c} 1 & 0 & \cdots & 0 \\ \hline 0 & \left(\frac{\lambda_1}{\lambda_0}\right)^k & \cdots & 0 \\ \hline \vdots & \ddots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & \left(\frac{\lambda_{m-1}}{\lambda_0}\right)^k \end{array}\right) y = X \left(\begin{array}{c|c|c|c} 1 & 0 & \cdots & 0 \\ \hline 0 & 0 & \cdots & 0 \\ \hline \vdots & \ddots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & 0 \end{array}\right) y \\ & = & X\psi_0 e_0 = \psi_0 X e_0 = \psi_0 x_0. \end{array}$$

Thus, as long as $\psi_0 \neq 0$ (which means v must have a component in the direction of x_0) this method will eventually yield a vector in the direction of x_0. However, this time the problem is that we don't know λ_0 when we start.

12.1.3 Convergence

Before we make the algorithm practical, let us examine how fast the iteration converges. This requires a few definitions regarding rates of convergence.

Definition 12.1 *Let* $\alpha_0, \alpha_1, \alpha_2, \ldots \in \mathbb{C}$ *be an infinite sequence of scalars. Then* α_k *is said to converge to* α *if*

$$\lim_{k\to\infty} |\alpha_k - \alpha| = 0.$$

Let $x_0, x_1, x_2, \ldots \in \mathbb{C}^m$ *be an infinite sequence of vectors. Then* x_k *is said to converge to* x *in the* $\|\cdot\|$ *norm if*

$$\lim_{k\to\infty} \|\alpha_k - \alpha\| = 0.$$

Notice that because of the equivalence of norms, if the sequence converges in one norm, it converges in all norms.

Definition 12.2 *Let* $\alpha_0, \alpha_1, \alpha_2, \ldots \in \mathbb{C}$ *be an infinite sequence of scalars that converges to* α*. Then*

- α_k *is said to converge linearly to* α *if for large enough* k
$$|\alpha_{k+1}-\alpha| \leq C|\alpha_k-\alpha|$$
for some constant $C<1$.
- α_k *is said to converge super-linearly to* α *if*
$$|\alpha_{k+1}-\alpha| \leq C_k|\alpha_k-\alpha|$$
with $C_k \to 0$.
- α_k *is said to converge quadratically to* α *if for large enough* k
$$|\alpha_{k+1}-\alpha| \leq C|\alpha_k-\alpha|^2$$
for some constant C.
- α_k *is said to converge super-quadratically to* α *if*
$$|\alpha_{k+1}-\alpha| \leq C_k|\alpha_k-\alpha|^2$$
with $C_k \to 0$.
- α_k *is said to converge cubically to* α *if for large enough* k
$$|\alpha_{k+1}-\alpha| \leq C|\alpha_k-\alpha|^3$$
for some constant C.

Linear convergence can be slow. Let's say that for $k \geq K$ we observe that
$$|\alpha_{k+1}-\alpha| \leq C|\alpha_k-\alpha|.$$
Then, clearly, $|\alpha_{k+n}-\alpha| \leq C^n|\alpha_k-\alpha|$. If $C=0.99$, progress may be very, very slow. If $|\alpha_k-\alpha|=1$, then
$$\begin{array}{rcl}
|\alpha_{k+1}-\alpha| & \leq & 0.99000 \\
|\alpha_{k+2}-\alpha| & \leq & 0.98010 \\
|\alpha_{k+3}-\alpha| & \leq & 0.97030 \\
|\alpha_{k+4}-\alpha| & \leq & 0.96060 \\
|\alpha_{k+5}-\alpha| & \leq & 0.95099 \\
|\alpha_{k+6}-\alpha| & \leq & 0.94148 \\
|\alpha_{k+7}-\alpha| & \leq & 0.93206 \\
|\alpha_{k+8}-\alpha| & \leq & 0.92274 \\
|\alpha_{k+9}-\alpha| & \leq & 0.91351
\end{array}$$
Quadratic convergence is fast. Now
$$\begin{array}{rcl}
|\alpha_{k+1}-\alpha| & \leq & C|\alpha_k-\alpha|^2 \\
|\alpha_{k+2}-\alpha| & \leq & C|\alpha_{k+1}-\alpha|^2 \leq C(C|\alpha_k-\alpha|^2)^2 = C^3|\alpha_k-\alpha|^4 \\
|\alpha_{k+3}-\alpha| & \leq & C|\alpha_{k+2}-\alpha|^2 \leq C(C^3|\alpha_k-\alpha|^4)^2 = C^7|\alpha_k-\alpha|^8 \\
& \vdots & \\
|\alpha_{k+n}-\alpha| & \leq & C^{2^n-1}|\alpha_k-\alpha|^{2^n}
\end{array}$$

Even $C = 0.99$ and $|\alpha_k - \alpha| = 1$, then

$$\begin{array}{rcl}
|\alpha_{k+1} - \alpha| & \leq & 0.99000 \\
|\alpha_{k+2} - \alpha| & \leq & 0.970299 \\
|\alpha_{k+3} - \alpha| & \leq & 0.932065 \\
|\alpha_{k+4} - \alpha| & \leq & 0.860058 \\
|\alpha_{k+5} - \alpha| & \leq & 0.732303 \\
|\alpha_{k+6} - \alpha| & \leq & 0.530905 \\
|\alpha_{k+7} - \alpha| & \leq & 0.279042 \\
|\alpha_{k+8} - \alpha| & \leq & 0.077085 \\
|\alpha_{k+9} - \alpha| & \leq & 0.005882 \\
|\alpha_{k+10} - \alpha| & \leq & 0.000034
\end{array}$$

If we consider α the correct result then, eventually, the number of correct digits roughly doubles in each iteration. This can be explained as follows: If $|\alpha_k - \alpha| < 1$, then the number of correct decimal digits is given by

$$-\log_{10} |\alpha_k - \alpha|.$$

Since $\log_{10}$ is a monotonically increasing function,

$$\log_{10} |\alpha_{k+1} - \alpha| \leq \log_{10} C|\alpha_k - \alpha|^2 = \log_{10}(C) + 2\log_{10} |\alpha_k - \alpha| \leq 2\log_{10}(|\alpha_k - \alpha|$$

and hence

$$\underbrace{-\log_{10} |\alpha_{k+1} - \alpha|}_{\substack{\text{number of correct} \\ \text{digits in } \alpha_{k+1}}} \geq 2(\underbrace{-\log_{10}(|\alpha_k - \alpha|}_{\substack{\text{number of correct} \\ \text{digits in } \alpha_k}}).$$

Cubic convergence is dizzyingly fast: Eventually the number of correct digits triples from one iteration to the next.

We now define a convenient norm.

Lemma 12.3 *Let $X \in \mathbb{C}^{m \times m}$ be nonsingular. Define $\| \cdot \|_X : \mathbb{C}^m \to \mathbb{R}$ by $\|y\|_X = \|Xy\|$ for some given norm $\| \cdot \| : \mathbb{C}^m \to \mathbb{R}$. Then $\| \cdot \|_X$ is a norm.*

Homework 12.4 *Prove Lemma 12.3.*

With this new norm, we can do our convergence analysis:

$$v^{(k)} - \psi_0 x_0 \;=\; A^k v^{(0)}/\lambda_0^k - \psi_0 x_0 = X \left(\begin{array}{c|c|c|c} 1 & 0 & \cdots & 0 \\ \hline 0 & \left(\frac{\lambda_1}{\lambda_0}\right)^k & \cdots & 0 \\ \hline \vdots & \ddots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & \left(\frac{\lambda_{m-1}}{\lambda_0}\right)^k \end{array} \right) X^{-1} v^{(0)} - \psi_0 x_0$$

$$= X \left(\begin{array}{c|c|c|c} 1 & 0 & \cdots & 0 \\ \hline 0 & \left(\frac{\lambda_1}{\lambda_0}\right)^k & \cdots & 0 \\ \hline \vdots & \ddots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & \left(\frac{\lambda_{m-1}}{\lambda_0}\right)^k \end{array} \right) y - \psi_0 x_0 = X \left(\begin{array}{c|c|c|c} 0 & 0 & \cdots & 0 \\ \hline 0 & \left(\frac{\lambda_1}{\lambda_0}\right)^k & \cdots & 0 \\ \hline \vdots & \ddots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & \left(\frac{\lambda_{m-1}}{\lambda_0}\right)^k \end{array} \right) y$$

Hence

$$X^{-1}(v^{(k)} - \psi_0 x_0) = \left(\begin{array}{c|c|c|c} 0 & 0 & \cdots & 0 \\ \hline 0 & \left(\frac{\lambda_1}{\lambda_0}\right)^k & \cdots & 0 \\ \hline \vdots & \ddots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & \left(\frac{\lambda_{m-1}}{\lambda_0}\right)^k \end{array} \right) y$$

and

$$X^{-1}(v^{(k+1)} - \psi_0 x_0) = \left(\begin{array}{c|c|c|c} 0 & 0 & \cdots & 0 \\ \hline 0 & \frac{\lambda_1}{\lambda_0} & \cdots & 0 \\ \hline \vdots & \ddots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & \frac{\lambda_{m-1}}{\lambda_0} \end{array} \right) X^{-1}(v^{(k)} - \psi_0 x_0).$$

Now, let $\| \cdot \|$ be a p-norm[1] and its induced matrix norm and $\| \cdot \|_{X^{-1}}$ as defined in Lemma 12.3. Then

$$\begin{aligned} \|v^{(k+1)} - \psi_0 x_0\|_{X^{-1}} &= \|X^{-1}(v^{(k+1)} - \psi_0 x_0)\| \\ &= \left\| \left(\begin{array}{c|c|c|c} 0 & 0 & \cdots & 0 \\ \hline 0 & \frac{\lambda_1}{\lambda_0} & \cdots & 0 \\ \hline \vdots & \ddots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & \frac{\lambda_{m-1}}{\lambda_0} \end{array} \right) X^{-1}(v^{(k)} - \psi_0 x_0) \right\| \\ &\leq \left| \frac{\lambda_1}{\lambda_0} \right| \|X^{-1}(v^{(k)} - \psi_0 x_0)\| = \left| \frac{\lambda_1}{\lambda_0} \right| \|v^{(k)} - \psi_0 x_0\|_{X^{-1}}. \end{aligned}$$

This shows that, in this norm, the convergence of $v^{(k)}$ to $\psi_0 x_0$ is linear: The difference between current approximation, $v^{(k)}$, and the solution, ψx_0, is reduced by at least a constant factor in each iteration.

12.1.4 Practical Power Method

The following algorithm, known as the Power Method, avoids the problem of $v^{(k)}$ growing or shrinking in length, without requiring λ_0 to be known, by scaling it to be of unit length at each step:

for $k = 0, \ldots$
  $v^{(k+1)} = Av^{(k)}$
  $v^{(k+1)} = v^{(k+1)} / \|v^{(k+1)}\|$
endfor

[1] We choose a p-norm to make sure that the norm of a diagonal matrix equals the absolute value of the largest element (in magnitude) on its diagonal.

12.1.5 The Rayleigh quotient

A question is how to extract an approximation of λ_0 given an approximation of x_0. The following theorem provides the answer:

Theorem 12.5 *If x is an eigenvector of A then $\lambda = x^H Ax/(x^H x)$ is the associated eigenvalue of A. This ratio is known as the Rayleigh quotient.*

Proof: Let x be an eigenvalue of A and λ the associated eigenvalue. Then $Ax = \lambda x$. Multiplying on the left by x^H yields $x^H Ax = \lambda x^H x$ which, since $x \neq 0$ means that $\lambda = x^H Ax/(x^H x)$.

Clearly this ratio as a function of x is continuous and hence an approximation to x_0 when plugged into this formula would yield an approximation to λ_0.

12.1.6 What if $|\lambda_0| \geq |\lambda_1|$?

Now, what if

$$|\lambda_0| = \cdots = |\lambda_{k-1}| > |\lambda_k| \geq \ldots \geq |\lambda_{m-1}|?$$

By extending the above analysis one can easily show that $v^{(k)}$ will converge to a vector in the subspace spanned by the eigenvectors associated with $\lambda_0, \ldots, \lambda_{k-1}$.

An important special case is when $k = 2$: if A is real valued then λ_0 still may be complex valued in which case $\bar{\lambda}_0$ is also an eigenvalue and it has the same magnitude as λ_0. We deduce that $v^{(k)}$ will always be in the space spanned by the eigenvectors corresponding to λ_0 and $\bar{\lambda}_0$.

12.2 The Inverse Power Method

The Power Method homes in on an eigenvector associated with the largest (in magnitude) eigenvalue. The Inverse Power Method homes in on an eigenvector associated with the smallest eigenvalue (in magnitude).

Throughout this section we will assume that a given matrix $A \in \mathbb{C}^{m\times m}$ is *nondeficient* and nonsingular so that there exist matrix X and diagonal matrix Λ such that $A = X\Lambda X^{-1}$. We further assume that $\Lambda = \mathrm{diag}(\lambda_0, \cdots, \lambda_{m-1})$ and

$$|\lambda_0| \geq |\lambda_1| \geq \cdots \geq |\lambda_{m-2}| > |\lambda_{m-1}|.$$

Theorem 12.6 *Let $A \in \mathbb{C}^{m\times m}$ be nonsingular. Then λ and x are an eigenvalue and associated eigenvector of A if and only if $1/\lambda$ and x are an eigenvalue and associated eigenvector of A^{-1}.*

Homework 12.7 *Assume that*

$$|\lambda_0| \geq |\lambda_1| \geq \cdots \geq |\lambda_{m-2}| > |\lambda_{m-1}| > 0.$$

Show that

$$\left|\frac{1}{\lambda_{m-1}}\right| > \left|\frac{1}{\lambda_{m-2}}\right| \geq \left|\frac{1}{\lambda_{m-3}}\right| \geq \cdots \geq \left|\frac{1}{\lambda_0}\right|.$$

Thus, an eigenvector associated with the smallest (in magnitude) eigenvalue of A is an eigenvector associated with the largest (in magnitude) eigenvalue of A^{-1}. This suggest the following naive iteration:

for $k = 0, \ldots$
  $v^{(k+1)} = A^{-1} v^{(k)}$
  $v^{(k+1)} = \lambda_{m-1} v^{(k+1)}$
endfor

Of course, we would want to factor $A = LU$ once and solve $L(Uv^{(k+1)}) = v^{(k)}$ rather than multiplying with A^{-1}. From the analysis of the convergence of the "second attempt" for a Power Method algorithm we conclude that now

$$\|v^{(k+1)} - \psi_{m-1} x_{m-1}\|_{X^{-1}} \le \left| \frac{\lambda_{m-1}}{\lambda_{m-2}} \right| \|v^{(k)} - \psi_{m-1} x_{m-1}\|_{X^{-1}}.$$

A practical Inverse Power Method algorithm is given by

for $k = 0, \ldots$
  $v^{(k+1)} = A^{-1} v^{(k)}$
  $v^{(k+1)} = v^{(k+1)} / \|v^{(k+1)}\|$
endfor

Often, we would expect the Invert Power Method to converge faster than the Power Method. For example, take the case where $|\lambda_k|$ are equally spaced between 0 and m: $|\lambda_k| = (k+1)$. Then

$$\left| \frac{\lambda_1}{\lambda_0} \right| = \frac{m-1}{m} \quad \text{and} \quad \left| \frac{\lambda_{m-1}}{\lambda_{m-2}} \right| = \frac{1}{2}.$$

which means that the Power Method converges much more slowly than the Inverse Power Method.

12.3 Rayleigh-quotient Iteration

The next observation is captured in the following lemma:

Lemma 12.8 *Let $A \in \mathbb{C}^{m \times m}$ and $\mu \in \mathbb{C}$. Then (λ, x) is an eigenpair of A if and only if $(\lambda - \mu, x)$ is an eigenpair of $(A - \mu I)$.*

Homework 12.9 *Prove Lemma 12.8.*

The matrix $A - \mu I$ is referred to as the matrix A that has been "shifted" by μ. What the lemma says is that shifting A by μ shifts the spectrum of A by μ:

Lemma 12.10 *Let $A \in \mathbb{C}^{m \times m}$, $A = X \Lambda X^{-1}$ and $\mu \in \mathbb{C}$. Then $A - \mu I = X(\Lambda - \mu I) X^{-1}$.*

Homework 12.11 *Prove Lemma 12.10.*

This suggests the following (naive) iteration: Pick a value μ close to λ_{m-1}. Iterate

for $k = 0, \ldots$
  $v^{(k+1)} = (A - \mu I)^{-1} v^{(k)}$
  $v^{(k+1)} = (\lambda_{m-1} - \mu) v^{(k+1)}$
endfor

Of course one would solve $(A - \mu I)v^{(k+1)} = v^{(k)}$ rather than computing and applying the inverse of A.

If we index the eigenvalues so that $|\lambda_0 - \mu| \le \cdots \le |\lambda_{m-2} - \mu| < |\lambda_{m-1} - \mu|$ then

$$\|v^{(k+1)} - \psi_{m-1} x_{m-1}\|_{X^{-1}} \le \left| \frac{\lambda_{m-1} - \mu}{\lambda_{m-2} - \mu} \right| \|v^{(k)} - \psi_{m-1} x_{m-1}\|_{X^{-1}}.$$

The closer to λ_{m-1} the "shift" (so named because it shifts the spectrum of A) is chosen, the more favorable the ratio that dictates convergence.

A more practical algorithm is given by

for $k = 0, \ldots$
  $v^{(k+1)} = (A - \mu I)^{-1} v^{(k)}$
  $v^{(k+1)} = v^{(k+1)} / \|v^{(k+1)}\|$
endfor

The question now becomes how to chose μ so that it is a good guess for λ_{m-1}. Often an application inherently supplies a reasonable approximation for the smallest eigenvalue or an eigenvalue of particular interest. However, we know that eventually $v^{(k)}$ becomes a good approximation for x_{m-1} and therefore the Rayleigh quotient gives us a way to find a good approximation for λ_{m-1}. This suggests the (naive) Rayleigh-quotient iteration:

for $k = 0, \ldots$
  $\mu_k = v^{(k)H} A v^{(k)} / (v^{(k)H} v^{(k)})$
  $v^{(k+1)} = (A - \mu_k I)^{-1} v^{(k)}$
  $v^{(k+1)} = (\lambda_{m-1} - \mu_k) v^{(k+1)}$
endfor

Now[2]

$$\|v^{(k+1)} - \psi_{m-1} x_{m-1}\|_{X^{-1}} \le \left| \frac{\lambda_{m-1} - \mu_k}{\lambda_{m-2} - \mu_k} \right| \|v^{(k)} - \psi_{m-1} x_{m-1}\|_{X^{-1}}$$

with

$$\lim_{k \to \infty} (\lambda_{m-1} - \mu_k) = 0$$

which means *super linear* convergence is observed. In fact, it can be shown that once k is large enough

$$\|v^{(k+1)} - \psi_{m-1} x_{m-1}\|_{X^{-1}} \le C \|v^{(k)} - \psi_{m-1} x_{m-1}\|^2_{X^{-1}},$$

which is known as quadratic convergence. Roughly speaking this means that every iteration doubles the number of correct digits in the current approximation. To prove this, one shows that $|\lambda_{m-1} - \mu_k| \le C \|v^{(k)} - \psi_{m-1} x_{m-1}\|_{X^{-1}}$.

[2] I think... I have not checked this thoroughly. But the general idea holds. λ_{m-1} has to be defined as the eigenvalue to which the method eventually converges.

Better yet, it can be shown that if A is Hermitian, then, once k is large enough,

$$\|v^{(k+1)} - \psi_{m-1}x_{m-1}\|_{X^{-1}} \leq C\|v^{(k)} - \psi_{m-1}x_{m-1}\|_{X^{-1}}^3,$$

which is known as cubic convergence. Roughly speaking this means that every iteration triples the number of correct digits in the current approximation. This is mind-boggling fast convergence!

A practical Rayleigh quotient iteration is given by

$v^{(0)} = v^{(0)}/\|v^{(0)}\|_2$
for $k = 0, \ldots$
  $\mu_k = v^{(k)H} A v^{(k)}$ (Now $\|v^{(k)}\|_2 = 1$)
  $v^{(k+1)} = (A - \mu_k I)^{-1} v^{(k)}$
  $v^{(k+1)} = v^{(k+1)}/\|v^{(k+1)}\|$
endfor

Bibliography

[1] E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, Jack J. Dongarra, J. Du Croz, S. Hammarling, A. Greenbaum, A. McKenney, and D. Sorensen. *LAPACK Users' guide (third ed.)*. SIAM, Philadelphia, PA, USA, 1999.

[2] E. Anderson, Z. Bai, J. Demmel, J. E. Dongarra, J. DuCroz, A. Greenbaum, S. Hammarling, A. E. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK Users' Guide*. SIAM, Philadelphia, 1992.

[3] Paolo Bientinesi. *Mechanical Derivation and Systematic Analysis of Correct Linear Algebra Algorithms*. PhD thesis, Department of Computer Sciences, The University of Texas, 2006. Technical Report TR-06-46. September 2006.
Download from http://www.cs.utexas.edu/users/flame/web/FLAMEPublications.html.

[4] Paolo Bientinesi, John A. Gunnels, Margaret E. Myers, Enrique S. Quintana-Ortí, and Robert A. van de Geijn. The science of deriving dense linear algebra algorithms. *ACM Trans. Math. Soft.*, 31(1):1–26, March 2005.
Download from http://www.cs.utexas.edu/users/flame/web/FLAMEPublications.html.

[5] Paolo Bientinesi, Enrique S. Quintana-Ortí, and Robert A. van de Geijn. Representing linear algebra algorithms in code: The FLAME APIs. *ACM Trans. Math. Soft.*, 31(1):27–59, March 2005.
Download from http://www.cs.utexas.edu/users/flame/web/FLAMEPublications.html.

[6] Paolo Bientinesi and Robert A. van de Geijn. The science of deriving stability analyses. FLAME Working Note #33. Technical Report AICES-2008-2, Aachen Institute for Computational Engineering Sciences, RWTH Aachen, November 2008.
Download from http://www.cs.utexas.edu/users/flame/web/FLAMEPublications.html.

[7] Paolo Bientinesi and Robert A. van de Geijn. Goal-oriented and modular stability analysis. *SIAM J. Matrix Anal. & Appl.*, 32(1):286–308, 2011.
Download from http://www.cs.utexas.edu/users/flame/web/FLAMEPublications.html.

[8] Paolo Bientinesi and Robert A. van de Geijn. Goal-oriented and modular stability analysis. *SIAM J. Matrix Anal. Appl.*, 32(1):286–308, March 2011.
Download from http://www.cs.utexas.edu/users/flame/web/FLAMEPublications.html.
We suggest you read FLAME Working Note #33 for more details.

[9] BLAST. ??? ???, ????

[10] I. S. Dhillon. *A New $O(n^2)$ Algorithm for the Symmetric Tridiagonal Eigenvalue/Eigenvector Problem*. PhD thesis, Computer Science Division, University of California, Berkeley, California, May 1997. Available as UC Berkeley Technical Report No. UCB//CSD-97-971.

[11] I. S. Dhillon. Reliable computation of the condition number of a tridiagonal matrix in $O(n)$ time. *SIAM J. Matrix Anal. Appl.*, 19(3):776–796, July 1998.

[12] I. S. Dhillon and B. N. Parlett. Multiple representations to compute orthogonal eigenvectors of symmetric tridiagonal matrices. *Lin. Alg. Appl.*, 387:1–28, August 2004.

[13] Inderjit S. Dhillon, Beresford N. Parlett, and Christof Vömel. The design and implementation of the MRRR algorithm. *ACM Trans. Math. Soft.*, 32(4):533–560, December 2006.

[14] J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart. *LINPACK Users' Guide*. SIAM, Philadelphia, 1979.

[15] Jack J. Dongarra, Jeremy Du Croz, Sven Hammarling, and Iain Duff. A set of level 3 basic linear algebra subprograms. *ACM Trans. Math. Soft.*, 16(1):1–17, March 1990.

[16] Jack J. Dongarra, Jeremy Du Croz, Sven Hammarling, and Richard J. Hanson. An extended set of FORTRAN basic linear algebra subprograms. *ACM Trans. Math. Soft.*, 14(1):1–17, March 1988.

[17] Jack J. Dongarra, Iain S. Duff, Danny C. Sorensen, and Henk A. van der Vorst. *Solving Linear Systems on Vector and Shared Memory Computers*. SIAM, Philadelphia, PA, 1991.

[18] Jack J. Dongarra, Sven J. Hammarling, and Danny C. Sorensen. Block reduction of matrices to condensed forms for eigenvalue computations. *Journal of Computational and Applied Mathematics*, 27, 1989.

[19] Robert A. van de Geijn Field G. Van Zee. BLIS: A framework for rapid instantiation of BLAS functionality. *ACM Trans. Math. Soft.*, 2015. To appear.

[20] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 3nd edition, 1996.

[21] Kazushige Goto and Robert van de Geijn. Anatomy of high-performance matrix multiplication. *ACM Trans. Math. Soft.*, 34(3):12:1–12:25, May 2008.
Download from http://www.cs.utexas.edu/users/flame/web/FLAMEPublications.html.

[22] John A. Gunnels. *A Systematic Approach to the Design and Analysis of Parallel Dense Linear Algebra Algorithms*. PhD thesis, Department of Computer Sciences, The University of Texas, December 2001.
Download from http://www.cs.utexas.edu/users/flame/web/FLAMEPublications.html.

[23] John A. Gunnels, Fred G. Gustavson, Greg M. Henry, and Robert A. van de Geijn. FLAME: Formal Linear Algebra Methods Environment. *ACM Trans. Math. Soft.*, 27(4):422–455, December 2001.
Download from http://www.cs.utexas.edu/users/flame/web/FLAMEPublications.html.

[24] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, second edition, 2002.

[25] C. G. J. Jacobi. Über ein leichtes Verfahren, die in der Theorie der Säkular-störungen vorkommenden Gleichungen numerisch aufzulösen. *Crelle's Journal*, 30:51–94, 1846.

[26] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh. Basic linear algebra subprograms for Fortran usage. *ACM Trans. Math. Soft.*, 5(3):308–323, Sept. 1979.

[27] Margaret E. Myers, Pierce M. van de Geijn, and Robert A. van de Geijn. *Linear Algebra: Foundations to Frontiers - Notes to LAFF With*. Self published, 2014.
Download from `http://www.ulaff.net`.

[28] G. W. Stewart. *Matrix Algorithms Volume 1: Basic Decompositions*. SIAM, Philadelphia, PA, USA, 1998.

[29] Robert van de Geijn and Kazushige Goto. *Encyclopedia of Parallel Computing*, chapter BLAS (Basic Linear Algebra Subprograms), pages Part 2, 157–164. Springer, 2011.

[30] Robert A. van de Geijn and Enrique S. Quintana-Ortí. *The Science of Programming Matrix Computations*. `www.lulu.com/contents/contents/1911788/`, 2008.

[31] Field G. Van Zee. `libflame`*: The Complete Reference*. `www.lulu.com`, 2012.
Download from http://www.cs.utexas.edu/users/flame/web/FLAMEPublications.html.

[32] Field G. Van Zee, Ernie Chan, Robert van de Geijn, Enrique S. Quintana-Ortí, and Gregorio Quintana-Ortí. The libflame library for dense matrix computations. *IEEE Computation in Science & Engineering*, 11(6):56–62, 2009.

[33] Field G. Van Zee, Robert A. van de Geijn, and Gregorio Quintana-Ortí. Restructuring the tridiagonal and bidiagonal qr algorithms for performance. 40(3):18:1–18:34, April 2014.
Download from http://www.cs.utexas.edu/users/flame/web/FLAMEPublications.html.

[34] Field G. Van Zee, Robert A. van de Geijn, Gregorio Quintana-Ortí, and G. Joseph Elizondo. Families of algorithms for reducing a matrix to condensed form. *ACM Trans. Math. Soft.*, 39(1), 2012.
Download from http://www.cs.utexas.edu/users/flame/web/FLAMEPublications.html.

[35] Stephen J. Wright. A collection of problems for which Gaussian elimination with partial pivoting is unstable. *SIAM J. Sci. Comput.*, 14(1):231–238, 1993.