Some notation:

Split the Frobenius norm into two pieces:

Let \( \| A \|_F^2 = S(A) + L(A) \)

\[
\begin{align*}
D(A) &= \sum_i |a_{ii}|^2 & \text{diagonal part} \\
L(A) &= \sum_{i \neq j} |a_{ij}|^2 & \text{off-diagonal part}
\end{align*}
\]

\[\Rightarrow S(A) = D(A) + L(A) = \| A \|_F^2.\]

**Theorem** Let \( A^{(k)} \) be the \( k \)th iterate in the Jacobi Algorithm. Then

\[
\lim_{k \to \infty} D(A^{(k)}) = \text{true}(A^*)
\]

**Proof:** Let \( a_{pq} \) be the off-diagonal element of \( A \) with the largest absolute value.

Let \( B = R^{(q)}(q)^T A R^{(q)}(q) \) (a single Jacobi Rotation)

\[
B_{pq} = (\cos q \sin q \sin q \cos q)(a_{pq} \quad 0) = 0 \cdot a_{pq} = 0 \text{ (by construction)}.
\]

But from the lemma, \( \| B \|_F^2 = \| R^T A R \|_F^2 = \| A \|_F^2.\)

\[\Rightarrow \text{This implies that } \quad b_{pp}^2 + b_{qq}^2 = a_{pp}^2 + a_{qq}^2 + 2a_{pq}^2.\]

Furthermore, \( S(B) = D(B) + L(B) \)

\[
= S(A) - S(A) = L(A).
\]
So this means that \( D(B) = D(A) + L(A) \)
\[ = D(A) + 2a_{pq}^2. \]
\[ \Rightarrow L(B) = L(A) - 2a_{pq}^2 = 0 \text{ for } A, B. \]

But, the same argument works for \( A \) and \( B \); the original non-nan matrix:
\[ S(B) = D(B) + L(B) \]
\[ = S(A) \]
\[ = D(A) + L(A) \]
\[ \Rightarrow L(B) = L(A) - 2a_{pq}^2. \]
\[ \Updownarrow \]
\[ \Rightarrow L(B) < L(A) \]

Continuing: Since \( a_{pq} \) was the largest off-diagonal element of \( A \), we have that \( L(A) \leq n(n-1)a_{pq} \)
\[ \Rightarrow a_{pq}^2 \geq \frac{L(A)}{n(n-1)} \]

Therefore, \( L(B) = L(A) - 2a_{pq}^2 \)
\[ \leq L(A) - 2 \frac{L(A)}{n(n-1)} \]
\[ = L(A) \left( 1 - \frac{2}{n(n-1)} \right) \]

Re-label the matrix:
\[ A^{(a)} = A \]
\[ A^{(b)} = B \]
\[ \Rightarrow L(A^{(a)}) \leq \left( 1 - \frac{2}{n(n-1)} \right) L(A^{(a)}) \]
\[ \Rightarrow L(A^{(b)}) \leq \left( 1 - \frac{2}{n(n-1)} \right) L(A^{(a)}) \leq \left( 1 - \frac{2}{n(n-1)} \right)^2 L(A^{(a)}). \]
After $k$ steps, \( L(A^{(k)}) = \left(1 - \frac{2}{n^{(m)}}\right)^k L(A^{(n)}) \)

So thereafter, \( L(A^{(k)}) \to 0 \) as \( k \to \infty \).

And since \( S(A^{(n)}) = D(A^{(n)}) + L(A^{(n)}) \)

\[ = \text{true}(A^2) \]

\[ \lim_{k \to \infty} \left(D(A^{(n)}) + L(A^{(n)})\right) = \lim_{k \to \infty} D(A^{(n)}) = \text{true}(A^2). \]

What guarantees that \( \lim_{k \to \infty} D(A^{(n)}) = \text{true}(A^2) \)

implies that \( A^{(n)} \to \begin{pmatrix} \lambda_1 & \cdots \ & \lambda_n \end{pmatrix} \) ?

**Idea** Apply Gerschgorin's Theorem:

\( A^{(n)} \) and \( A \) have the same eigenvalues, and \( L(A^{(n)}) = 0 \).

Therefore the Gerschgorin disks of \( A^{(k)} \) have radii going to 0 as well. Therefore, the eigenvalues of \( A^{(k)} \), and therefore \( A \), are the limit of the diagonal of \( A^{(k)} \).

What about the rate of convergence?

We showed that \( L(A^{(n)}) = \left(1 - \frac{2}{n^{(m)}}\right)^k L(A^{(n)}) \)

if \( n = 1000 \), \( 1 - \frac{2}{n^{(m)}} = 0.99999799799... \)

A if \( k = 100 \), \( \left(1 - \frac{2}{n^{(m)}}\right)^k \approx 0.997997 \)

\( k = 10000 \), \( \left(1 - \frac{2}{n^{(m)}}\right)^k \approx 0.98 \)

Real-life convergence is often much faster than indicated in the proof.
One final note:

Jacobi's Algorithm can be terminated when $L(A^{(n)}) < \epsilon$

$A^{(n)} = \begin{align*}
\overbrace{R_{(k-1)}^T \cdots R_{(q)}^T \cdots R_{(q)}^T}^{R^T} \underbrace{A \cdots A}_{R} \overbrace{R_{(q)}^T \cdots R_{(q)}^T}^{R}
\end{align*}$

$\approx$ diagonal

$A = R A^{(n)} R^T$

$\approx$ diagonal

$R$ diagonalizes $A$.

$\Rightarrow R$ is the matrix of eigenvectors of $A$ (approximate eigenvectors).

$\Rightarrow A^{(n)}$ has eigenvalues (approximation of)

This means that Jacobi algorithm computes all eigenvalues and eigenvectors at the same time.

The QR Method

For general matrices, the QR method can be used to find all eigenvalues. Examine algorithm, then analyze:

Algorithm

$A^{(0)} = A$

For $k = 1, 2, ...$

Factor $A^{(k-1)} = Q^{(k)} R^{(k)}$

Compute $A^{(k)} = R^{(k)} Q^{(k)}$

Under certain assumptions,

$A^{(k)} \rightarrow$ upper triangular,

and since $R^{(k)} = Q^{(k)} A^{(k-1)}$

$\Rightarrow A^{(k)} = Q^{(k)} A^{(k-1)} Q^{(k)}$

so $A^{(k)}$ and $A$ have the same eigenvalues.

[4]
Three things must be done in order for this to be a usable, accelerated algorithm:

1. First reduce A to tridiagonal, or Hessenberg, form using Householder reflections.
2. Apply to shifted matrix, \( A^{(k)} - \mu^{(k)} I \), with \( \mu^{(k)} \) an estimate for some \( \lambda \).
3. Use deflation - decoupling into smaller problems.

To understand the "pure" QR algorithm above, first study subspace iteration (basically power method on groups of vectors):

**Subspace Iteration**

Pick \( Q^{(0)} \in \mathbb{R}^{m \times m} \), orthonormal.

For \( k = 1, 2, ... \):

Set \( z = A Q^{(k-1)} \).

Factor \( z = Q^{(k)} R^{(k)} \).

Then \( Q^{(k)} \) is top eigenvalue of \( A \) at a rate of

\[
\max_{\text{top, min}} \left( \frac{\text{eigenvalue}}{x_p} \right)^k.
\]

More details later...