CME302 class notes

Erich Trieschman

2021 Fall quarter

1 Linear algebra review

1.1 Vector products

The **inner product**, also known as the dot product, results in a scalar

- $x^T y = \sum x_i * y_i$; $x^T y = \|x\|_2 \|y\|_2 \cos\theta$; $x^T y = 0 \Leftrightarrow x \perp y$

The **outer product** results in a matrix. It is the outer sum of the two vectors, which can be of different lengths.

1.2 Norms

All norms, matrix or vector, satisfy

- Only zero vector has zero norm: $\|x\|_x = 0 \Leftrightarrow x = 0$
- $\|\alpha x\|_x = |\alpha| \|x\|_x$
- $\|x + y\|_x \leq \|x\|_x + \|y\|_x$ (Triangle inequality I), $\|x - y\|_x \geq \|x\|_x - \|y\|_x$ (Triangle inequality II)

1.2.1 Vector norms

Types of **vector norms**, $x \in \mathbb{R}^n$ (norm selection can give you solutions with different properties)

- $\|x\|_1 = \sum_{i=1}^n |x_i|$; $\|x\|_2 = \sqrt{\sum_{i=1}^n (x_i)^2}$; $\|x\|_\infty = \max_{i \in i,\ldots,n} |x_i|$; $\|x\|_p = (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}$

Cauchy-Schwarts Inequality: $|x^T y| \leq \|x\|_2 \|y\|_2$ (note equality when $x^T y = 0$)

Holder's Inequality: $|x^T y| \leq \|x\|_p \|y\|_q$, for p, q , s.t. $\frac{1}{p} + \frac{1}{q} = 1$

1.2.2 Matrix norms

Types of **matrix norms**, $A \in \mathbb{R}^{n \times m}$

- $\|A\|_\infty = \sup_{x \neq 0} \frac{\|Ax\|_\infty}{\|x\|_\infty} = \max_{\|x\|_\infty = 1} \|Ax\|_\infty = \max_i \left\|a_i^T\right\|_1$
- $\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p} = \max_{\|x\|_p = 1} \|Ax\|_p$
- $\|A\|_F = \sqrt{\sum_{i,j} a_{ij}^2} = \sqrt{tr(AA^T)} = \sqrt{tr(A^T A)} = \sqrt{\sum_{k=1}^{min(m,n)} \sigma_k^2}$

Submultiplicative inverse: $\|AB\|_p \leq \|A\|_p \|B\|_p$. Note: this is not always true for Frobenius norms.

Induced p-norm: $\|Ay\|_p \leq \|A\|_p \|y\|_p$

Orthogonally invariant: Orthogonal matrices do not change the norms of vectors or matrices:

- $\|Qx\|_x = \|x\|_x$; $\|QA\|_x = \|A\|_x, x \in \{p, F\}$

Other norm properties:

- $\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n} \|x\|_\infty$; $\|A\|_2 \leq \sqrt{m} \|A\|_\infty$; $\|A\|_\infty \leq \sqrt{n} \|A\|_2$

1.3 Matrix properties

1.3.1 Determinant

The **determinant** represents how the volume of a hypercube is transformed by the matrix.

- For square matrix, $det(\alpha A) = \alpha^n det(A)$; $det(AB) = det(A)det(B)$
- $det(A) = det(A^T)$; $det(A^{-1}) = \frac{1}{det(A)}$
- For square matrix, A singular $\Leftrightarrow det(A) = 0 \Leftrightarrow$ columns of A are not linearly independent

1.3.2 Trace

The trace of a matrix $A \in \mathbb{R}^{mxn}, tr(A)$, is equal to the sum of the entries in its diagonal, $tr(A) = \sum_{i=1}^{n} a_{ii}$. And a few properties of the trace:

- $tr(A) = tr(A^T)$; $tr(A + \alpha B) = tr(A) + \alpha tr(B)$; For two vectors, $u, v \in \mathbb{R}, tr(uv^T) = v^T u$
- Trace is invariant under cyclic permutations, that is $tr(ABCD) = tr(BCDA) = tr(CDAB) = tr(DABC)$

1.3.3 Inverses and transposes

The inverse of the transpose is the transpose of the inverse:

- $A^T(A^{-1})^T = (A^{-1}A)^T = I^T = I$
- $(A^{-1})^T A^T = (AA^{-1})^T = I^T = I$

1.3.4 Sherman-Morrison-Woodbury formula

for $A \in \mathbb{R}^{n\times n}, U, V \in \mathbb{R}^{n\times k}$

$$(A + UV^T)^{-1} = A^{-1} - A^{-1}U(I + V^T A^{-1}U)^{-1}V^T A^{-1}$$

Proof: begin with the inverse of the LHS multiplied by the RHS: $(A+UV^T)(A^{-1} - A^{-1}U(I+V^T A^{-1}U)^{-1}V^T A^{-1})$. Next perform matrix multiplication. The end result will be I, implying that the RHS is an inverse of $(A + UV^T)$

1.4 Orthogonal matrices

An orthogonal matrix, Q is a matrix whose columns are orthonormal. That is, $q_i^T q_j = 1$ for $i = j$, and $q_i^T q_j = 0$ for $i \neq j$. Equivalently, $Q^T Q = I$. For square matrices, $Q^T Q = QQ^T = I$

1.5 Projections, reflections, and rotations

1.5.1 Projections

A projection, v, of vector x onto vector y can be written in the form $v = \frac{y^T x}{y^T y} y$. **Projection matrices** are square matrices, P, s.t., $P^2 = P$.

1.5.2 Reflection

- P is a reflection matrix $\Leftrightarrow P^2 = I$
- P can be written in the form $P = I - \beta vv^T$, with $\beta = \frac{2}{v^T v}$, and v the vector orthogonal to the line/plane of reflection
- It can be shown that $Px = x \Leftrightarrow v^T x = 0$. These x are called the "fixed points" of P

1.6 Symmetric Positive Definite (SPD) Matrices

For A**, SPD,** i) $A = A^T$, ii) $x^T Ax > 0\ \forall x \neq 0$, iii) $a_{ii} > 0$, iv) $\lambda(A) \geq 0$, v) for B nonsingular, $B^T AB$ is also SPD.

When proving properties of SPDs, use the **following tricks:** i) Multiply by e_i since $e_i \neq 0$, ii) Use matrix transpose property, $x^T A^T = (Ax)^T$ to rearrange formulas

1.6.1 $B^T AB$ is also SPD

If A SPD $\Rightarrow B^T AB$ SPD for B nonsingular:

$$x^T B^T ABx = (Bx)^T A(Bx) > 0, (\text{since } B \text{ nonsingular} \Rightarrow Bx \neq 0)$$

1.7 Eigenvalues

Observe by definition $Ax = \lambda x \longleftrightarrow Ax - \lambda x = 0 \longleftrightarrow (A - \lambda I)x = 0$. To find lambda, we solve for the system of equations to satisfy $(A - \lambda I)x = 0$

The **algebraic multiplicity** of an eigenvalue, λ_i, is the number of times that λ_1 appears in $\lambda(A)$
The **geometric multiplicity** of an eigenvalue, λ_i, is the dimension of the space spanned by the eigenvectors of λ_i

Other **eigenvalue properties:** $\lambda(A) = \lambda(A^T)$; Courant-Fischer minmax theorem: $\lambda_1 = \max_{x \neq 0} \frac{x^T Ax}{\|x\|_2^2}$

1.7.1 Determinants and trace

$$det(A) = \prod_{i=1}^{n} \lambda_i \qquad\qquad tr(A) = \sum_{i=1}^{n} \lambda_i$$

1.7.2 Triangular matrices

For T triangular, the eigenvalues appear on the diagonal: $t_{ii} = \lambda_i, \forall i \in \{1, \ldots, n\}$
Corollary: T nonsingular $\Leftrightarrow$ all $t_{ii} \neq 0$

1.7.3 Gershgorin disc theorem

Gershgorin disc, $\mathbb{D}_i$, defined

$$\mathbb{D}_i = \{z \in \mathbb{C} \mid |z - a_{ii}| \leq \sum_{j \neq i} |a_{ij}|\}$$

All eigenvalues of A, $\lambda(A) \in \mathbb{C}$ are located in one of its Gershgorin discs. **Proof:**

$$\begin{aligned}
&Ax = \lambda x \longleftrightarrow (A - \lambda I)x = 0 \longleftrightarrow \sum_{j \neq i} a_{ij}x_j + (a_{ii} - \lambda)x_i = 0,\ \forall i \in \{1, \ldots, n\} \\
&\text{Choose } i\ s.t. |x_i| = \max_i |x_i| \\
&\qquad |(a_{ii} - \lambda)| = |\sum_{j \neq i} \frac{a_{ij}x_j}{x_i}| \leq \sum_{j \neq i} |\frac{a_{ij}x_j}{x_i}| \text{ , by triangle inequality} \\
&\qquad |(\lambda - a_{ii})| \leq \sum_{j \neq i} |a_{ij}|, \text{ since } |\frac{x_j}{x_i}| \leq 1
\end{aligned}$$

2 Matrix Decompositions

2.1 Schur Decomposition

For any $A \in \mathbb{C}^{n \times n}$, $A = QTQ^H$, where Q unitary $(Q^H Q = I), Q \in \mathbb{C}^{n \times n}$, T upper triangular

When $A \in \mathbb{R}^{n \times n}$, $A = QTQ^T$, where Q orthogonal $(Q^T Q = I), Q \in \mathbb{R}^{n \times n}$, T upper triangular

Note: If T is relaxed from strict upper triangular to block upper triangular (blocks of 2×2 or 1×1 on the diagonal), then Q can be selected to be in $\mathbb{R}^{n \times n}$.

2.2 Eigenvalue Decomposition

For A diagonalizable ($A \in \mathbb{R}^{n \times n}$ with n linearly independent eigenvectors), it can be decomposed as

$$A = X \Lambda X^{-1}, \text{ where } \Lambda \text{ a diagonal matrix of the eigenvalues of } A$$

For A real symmetric, A can be decomposed as $A = Q \Lambda Q^T, Q$ orthogonal

For A unitarily diagonalizable ($\Leftrightarrow$ normal: $A^H A = AA^H$), $A = Q \Lambda Q^H, Q$ unitary. When A complex Hermitian $(A = A^H)$, $\Lambda \in \mathbb{R}$

2.3 Singular Value Decomposition

Definition: For any $A \in \mathbb{C}^{m\times n}$ there exist two unitary matrices, $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$, and a diagonal matrix $\Sigma \in \mathbb{R}^{m\times n}$ such that $A = U\Sigma V^H$. When $A \in \mathbb{R}^{m\times n}$, $A = U\Sigma V^T$ with $U, V, \Sigma \in \mathbb{R}$

The singular values, σ_i of Σ are always ≥ 0. And by convention, they're ordered in decreasing order, so $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$

Derivation: Observe A^TA symmetric: $(A^TA)^T = A^TA$

$$\begin{aligned}
&A^TA \text{ symmetric} \Rightarrow \exists\, Q \text{ orthogonal and } \Lambda \text{ diagonal matrix of } \lambda_i \text{ s.t.,}\\
A^TA &= Q\Lambda Q^T\\
Q^TA^TAQ &= Q^TQ\Lambda Q^TQ\\
(AQ)^T(AQ) &= \Lambda, \text{ note } AQ \text{ is orthogonal, but not scaled to 1. Instead, each row is}\\
&\quad\text{scaled to the eigenvalue in that row: } \lambda_i = \|Aq_i\|_2^2
\end{aligned}$$

When A is full rank,

$$\begin{aligned}
A &= AQQ^T\\
&= (AQ)Q^T\\
&= AQD^{-1}DQ^T, \text{ where } D = \begin{bmatrix} \sqrt{\lambda_1} & \dots & 0\\ \vdots & & \vdots\\ 0 & \dots & \sqrt{\lambda_n}\end{bmatrix} \text{ and } D^{-1} = \begin{bmatrix} \frac{1}{\sqrt{\lambda_1}} & \dots & 0\\ \vdots & & \vdots\\ 0 & \dots & \frac{1}{\sqrt{\lambda_n}}\end{bmatrix}\\
A &= U\Sigma V^T, \text{ where } U = AQD^{-1}, \Sigma = D, V^T = Q^T
\end{aligned}$$

When A is not full rank, make the tall/thin SVD

And a few properties and remarks of $A \in \mathbb{R}^{n\times m}$ SVD

- $\|A\|_2 = \sigma_1$; $\left\|A^{-1}\right\|_2 = \frac{1}{\sigma_n}$ when A nonsingular; $\|A\|_F = \sqrt{\sum_i^{min\{n,m\}} \sigma_i^2}$; **Condition number**, $\kappa(A) = \|A\|_2 \left\|A^{-1}\right\|_2 = \frac{\sigma_1}{\sigma_n}$
- When A symmetric, $\sigma_i = |\lambda_i|$; When A orthogonal, $\sigma_1 = \cdots = \sigma_n = 1$
- The eigenvalues of A^TA and AA^T are the squares of the singular values of A, $\sigma_1^2, \dots, \sigma_n^2$
- By construction, V contains the eigenvectors of A^TA and U contains the eigenvectors of AA^T, so $A^TAv_i = \sigma_i^2 v_i$ and $AA^Tu_i = \sigma_i^2 u_i$

3 Error analysis

3.1 Floating point arithmetic

$$\pm\left(\sum_{i=1}^{t-1} d_i \beta^{-i}\right)\beta^e$$

Where β is the base (in floating point computation, $\beta = 2$), $d_0 \geq 1$, and $d_i \leq \beta - 1$, e is called the **exponent**, this is the location of the decimal place, $t-1$ in the summand is called the **precision** and indicates the number of digits (in base β) that can be stored with the number.

3.2 Unit roundoff

The **unit roundoff** for a floating-point number is

$$u = \frac{1}{2} \times \beta^{-(t-1)} \text{ (distance between the smallest digits stored in a floating-point number)}$$

For double precision floating point numbers (64 bits), $u \approx 10^{-16}$. The relative sensitivity of a problem is often called the **conditioning** of the problem

- Sensitivity: $\frac{\left\|\tilde{f}(x)-f(x)\right\|_p}{\|\tilde{x}-x\|_p}$; Relative sensitivity: $\frac{\left\|\tilde{f}(x)-f(x)\right\|_p \|x\|_p}{\|\tilde{x}-x\|_p \|f(x)\|_p}$

4 LU Factorization

The LU factorization makes it computationally easier to solve linear equations If we can decompose a matrix, A, into a product of a lower triangular matrix, L, and an upper triangular matrix, U, then to solve $Ax = b$, we can start by solving $Lz = b$, and then $Ux = z$. x, here, is the solution!

4.1 Basic algorithm

- Construct u_1^T equal to the first row of A, a_1^T
- Construct l_1 equal to each of the elements in the first column of A, a_1, divided by a_{11}, the "pivot"
- Calculate $A' \leftarrow A - l_1 u_1^T$. In practice (and somewhat confusingly), A' is now referred to as A
- Repeat the algorithm with the updated A, and the next row/column. Observe each l_i, u_i^T constructed are the rows/columns of the lower and upper triangular matrices of L, U respectively.

4.1.1 Gauss transforms

To compute $A = LU$, consider $L^{-1}A = U$, with L^{-1} that "zeros-out" the columns of A to get U. Call L^{-1}, G. As with the iterative algorithm above, we can multiply A by iterative G_i's to get U:

$$L^{-1}A = G_n G_{n-1} \ldots G_2 G_1 A = U \longrightarrow A = G_1^{-1} \ldots G_n^{-1} U = LU$$

4.2 Pivoting

4.2.1 When pivoting is needed

Notice that this algorithm relies on the pivots, a_{kk}, being nonzero. It turns out this will occur if none of the $k \times k$ blocks of A, $A[1:k, 1:k]$, have a determinant of 0. **Proof by induction**:
Case k=1:

$$A_1 = L_1 U_1 \longleftrightarrow det(A_1) = det(L_1 U_1) \longleftrightarrow det(A_1) = det(L_1) det(U_1), \text{ by property of determinants}$$
$$det(A_1) = det(U_1), \text{ since determinant of a triangular matrix is a product of the diagonals and the diagonal of } L_1 \text{ are 1's}$$
$$det(A_1) = a_{11} = u_{11} \rightarrow \text{so when determinant is not zero, we have a nonzero pivot}$$

4.3 Cholesky factorization

The Cholesky factorization is an LU factorization for Symmetric Positive Definite (SPD) matrices, where SPD matrix, $A = GG^T$, with G lower triangular.

4.4 Schur complement

A useful way to think about the LU factorization is with the **Schur complement** matrix structure. First observe A can be written in the following form

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

If we run the LU factorization algorithm for k steps, the resulting $A' = A$ is equal to

$$A = \begin{bmatrix} I & 0 \\ A_{21}A_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} - A_{21}A_{11}^{-1}A_{12} \end{bmatrix} \begin{bmatrix} I & A_{21}A_{11}^{-1} \\ 0 & I \end{bmatrix}$$

The bottom-right block of $A' = A, A'_{22} = A_{22}$ is equal to $A_{22} - A_{21}A_{11}^{-1}A_{12}$ from the original matrix. This is called the **Schur complement** of A

4.4.1 Schur complement derivation

At any step in the LU factorization, A can be written in the form

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix}$$

From this equality, we can create a system of equations and derive the Schur complement

5 QR factorization

The QR factorization decomposes a matrix, $A \in \mathbb{R}^{m \times n}, m \geq n$ into an orthogonal (orthonormal) matrix, Q, and an upper triangular matrix, R. When $A \in \mathbb{C}^{m \times n}$, Q is unitary.

Recall for $Q \in \mathbb{R}$, orthogonal, $Q^T Q = I$; for $Q \in \mathbb{C}$, unitary, $Q^H Q = I$; $\|Qx\|_2 = \|x\|_2$

If A is skinny (i.e., $n << m$), QR can take two different forms. $Q \in \mathbb{R}^{m \times m}$ can be square and $R \in \mathbb{R}^{m \times n}$ can be skinny. Or $Q \in \mathbb{R}^{m \times n}$ can be skinny and $R \in \mathbb{R}^{n \times n}$ can be square.

5.1 The QR factorization is unique

Proof that the QR factorization is unique for full rank matrix, A:

$$A = QR \longleftrightarrow Q^T A = R \longleftrightarrow^T Q^T A = R^T R \longleftrightarrow (QR)^T A = R^T R \longleftrightarrow A^T A = R^T R$$

We now have a matrix, $A^T A$ that can be written of the form $R^T R$, which is the structure of the Cholesky factorization. Suffice to show that $A^T A$ is Symmetric and Positive Definite (SPD) to prove the uniqueness of R.

5.2 Householder reflection

- Construct Q^T for each column in A that projects it onto a corresponding column of an upper right triangular matrix, R.
- E.g., for first column a_1: Want Q_1^T such that $Q_1^T a_1 = r_1$, where $r_1 = \pm \|a_1\|_2 e_1$ (since Q^T is orthogonal). This equates to finding Q_1^T that reflects a_1 onto e_1
- **The key** to the iterative part of the algorithm is to construct $Q_i^T, i > 1$ with an identity matrix in the upper-left $i-1 \times i-1$ quadrant, and a smaller Q_i^{*T} in the lower right $n-i \times n-i$ quadrant, filling the remaining sections of the matrix with 0's.

The **Householder reflection** maps $a \rightarrow \|a\|_2 e_1$ with

$$P = I - \beta v v^T, \text{ where } v = a - \|a\|_2 e_1, \text{ and } \beta = 2/v^T v$$

Aside: The fixed points of a reflection, P, remain unchanged when multiplied by the reflection, $Px = x$. Geometrically, these are the points that are *orthogonal* to the vector v defining the reflection (i.e., $v^T x = 0$)

5.3 Givens transformation

5.3.1 Givens transformation algorithm

A **Givens rotation** rotates $u = (u_1, u_2)^T$ to $\|u\|_2 e_1$. The matrix that does this, G^T, is defined by

$$G^T = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}, c = \frac{u_1}{\|u\|_2}, s = -\frac{u_2}{\|u\|_2}$$

Sequentially, the P_i's can multiply A to arrive at R

5.4 Gram-Schmidt transformation

Construction of r_{kk}, q_k, r_{kj}

$$a_k = \sum_{i=1}^{k} r_{ik} q_i = r_{kk} q_k + \sum_{i=1}^{k-1} r_{ik} q_i$$

1. $r_{ik} = q_i^T a_k$ for each $r_{ik}, i < k$, since Q orthonormal and q_{k-1} known
2. $z = r_{kk} q_k = q_k - \sum_{i=1}^{k-1} r_{ik} q_i$
3. $r_{kk} = \|z\|_2, \; q_k = \frac{z}{r_{kk}}$

5.5 QR factorization to solve least-squares problems

When A is tall and thin, it is unlikely that we get a solution to $Ax = b$. Instead, we choose to solve the least-squares problem, $argmin_x \|Ax - b\|_2$.

5.5.1 Method of normal equations

$$\text{Want: } (b - Ax) \perp \{z|z = Ay\} \longleftrightarrow (b - Ax) \perp range(A) \longleftrightarrow (b - Ax) \perp a_i, \forall i \in A$$
$$a_1^T(b - Ax) = 0, \forall i \in A \longleftrightarrow A^T(b - Ax) = 0 \longleftrightarrow x = (A^T A)^{-1}A^T b$$

5.5.2 QR method for least squares

$$A^T(Ax - b) = 0 \longleftrightarrow R^T Q^T(Ax - b) = 0$$
$$Q^T(Ax - b) = 0, \text{ since we assume } A, R \text{ full rank (multiply both sides by } R^{-T})$$
$$Q^T QRx - Q^T b = 0 \longleftrightarrow Rx = Q^T b \longleftrightarrow x = R^{-1}Q^T b$$

5.5.3 SVD for rank-deficient A

When A not full rank, we add constraint $\min_x \|x\|_2$. We can use the "thin" version of the Singular Value Decomposition to solve this

$$(Ax - b) \perp range(A) \longleftrightarrow (Ax - b) \perp range(U), \text{ since } R(A) = R(U) \text{ for } A = U\Sigma V^T$$
$$U^T(Ax - b) = 0 \longleftrightarrow U^T(U\Sigma V^T x - b) = 0 \longleftrightarrow \Sigma V^T x = U^T b$$
$$x = V\Sigma^{-1}U^T b \text{ (the "thin" SVD here provides a nonsingular } \Sigma \in \mathbb{R}^{r\times r}\text{, so we can take the inverse}$$

Observe for $\min_x \|x\|_2$ that the $x \perp N(A)$ is the shortest vector between $N(A)$ and the vector/plane of solutions to $argmin_x \|Ax - b\|_2$. This value must be in $R(V)$ since $R(V) = N(A)^\perp$

6 Iterative methods to find eigenvalues

6.1 Power iteration

Given $\lambda_1 > \lambda_2 \geq \cdots \geq \lambda_n \in \lambda(A)$, the **Power iteration** finds λ_1. This process assumes A is diagonalizable

$$A^k = \sum_{i=1}^{n} \lambda_i^k x_i y_i^T \text{ where } Y = X^{-1}$$
$$A^k \approx \lambda_1^k x_1 y_1^T \text{ since } \lambda_1 > \lambda_2$$
$$A^k q \approx \lambda_1^k x_1 y_1^T q = \lambda_1^k(y_1^T q)x_1, \text{ since } y_1^T q \text{ is a scalar. Observe } A^k q \parallel x_1$$

This theory is implemented in practice with the following formula

1. q_0, vector chosen at random
2. $z_k = Aq_k = A^k q_0$, evaluating for convergence if $z_k \parallel q_k \to z_k^T x_k = \|z\|_2 \|x\|_2$
3. $q_{k+1} = \dfrac{z_k}{\|z_k\|_2} = \dfrac{A^k q_0}{\|A^k q_0\|_2} \approx (\dfrac{\lambda_2}{|\lambda_1|})^k x_1$

Since $A^k q_0 = Aq_k \approx \lambda_1 x_1$, where $\|x_1\|_2 = 1$ ($WLOG$) and $q_k \parallel x_1$, we can solve for λ:

$$Aq_k \approx \lambda_1 x_1 \Longrightarrow Ax_1 \approx \lambda_1 x_1 \Rightarrow x_1^H Ax_1 \approx \lambda_1$$

Convergence: $O(|\frac{\lambda_1}{\lambda_2}|)^K)$, since

$$A^k q_0 = \sum_i \alpha_i A^k x_i = \sum_1 \alpha_i \lambda_i^k x_i = \alpha_1 \lambda_1^k (x_i + \frac{\alpha_2}{\alpha_1}(\frac{\lambda_2}{\lambda_1})^k + \cdots + \frac{\alpha_n}{\alpha_1}(\frac{\lambda_n}{\lambda_1})^k) \Longrightarrow \|A^k q_0\|_2 = |\alpha_1 \lambda_1^k|(1 + O(\frac{\lambda_2}{\lambda_1})^k)$$

6.2 Inverse iteration

Get the eigenvector for the eigenvalue closest to μ. Observe $(A - \mu I)^{-1}$ has the same eigenvectors of A:

$$(A - \mu I)^{-1}x = \lambda x \longleftrightarrow x = (A - \mu I)x = \lambda Ax - \lambda\mu x \longleftrightarrow \lambda Ax = x + \lambda\mu x \longleftrightarrow Ax = \frac{(1 + \lambda\mu)}{\lambda}x$$

Performing the power iteration on $(A - \mu I)^{-1}$, the largest eigenvalue to emerge will be of the form $\frac{1}{\lambda_i - \mu}$, and we get

$$(A - \mu I)^{-1k} q_0 = (A - \mu I)^{-1} q_k \approx \lambda_i x_i, \text{ where } \|x_i\|_2 = 1 \ (WLOG) \text{ and } q_k \parallel x_i$$

Since x_i is also an eigenvalue of A, we can solve $x_i^H Ax_i = \lambda_i$ for the λ_i closest in magnitude to μ.
Convergence: $O(|\frac{\lambda_i - \mu}{\lambda_j - \mu}|)^k)$, where λ_j is the next closest eigenvalue to μ

6.3 Eigenvalues of similar matrices

Theorem: For S nonsingular and $A = S^{-1}BS$, then i) $\lambda(A) = \lambda(B)$ and ii) x eigenvector of $A \Leftrightarrow S^{-1}x$ eigenvector of B.

6.4 Eigenvalues from invariant subspaces

Theorem: $X \in \mathbb{R}^{n\times m}$ is an invariant subspace of $A \in \mathbb{R}^{n\times n} \Leftrightarrow$ there is a $B \in \mathbb{R}^{n\times m}$ such that $AX = XB$. **Proof:**

$$\Rightarrow: X \text{ invariant} \longrightarrow Ax_i \in X \longrightarrow Ax_i = \sum_{j=1}^{m} x_j b_{ji} \longrightarrow AX = XB$$

Furthermore, when $AX = XB$, the m eigenvalues of B are also eigenvalues of A: $By = \lambda y \longrightarrow XBy = \lambda Xy \longrightarrow AXy = \lambda Xy$

6.5 Orthogonal iteration

First, consider how to construct orthogonal columns to reveal subsequent eigenvalues. Assume we use power iteration to compute q_1

$$\begin{aligned}
A^k &= \lambda_1 x_1 y_1^T + \lambda_2 x_2 y_2^T + \dots \\
PA^k &= \lambda_1 P x_1 y_1^T + \lambda_2 P x_2 y_2^T + \dots, \text{ where } P = I - x_1 x_1^T \\
PA^k &= 0 + \lambda_2 P x_2 y_2^T + \dots, \text{ since } Px_1 = Ix_1 - x_1 x_1^T x_1 = x_1 - x_1 = 0
\end{aligned}$$

PA can now be used to apply the power iteration to to reveal λ_2 and $(I - x_1^T x_1)x_2$

The general process is: build P_r, orthogonal projector onto $\{q_1, \dots, q_{r-1}\}^\perp$, use power iteration to reveal (λ_r, q_r)
Now consider the QR decomposition of X, observing its connection to the Schur Decomposition:

$$A = X\Lambda X^{-1} = QR\Lambda R^{-1}Q^H = QTQ^H, \text{ where upper triangular } T = R\Lambda R^{-1}$$

- The eigenvalues of A are on the diagonal of T
- By construction, each column of Q is projecting the corresponding column of X onto a vector orthogonal to the preceding ones
- The span of the columns of Q, $span\{q_1, \dots, q_n\}$ will be equal to the span of the columns of X, $span\{x_1, \dots, x_n\}$.

The process for the **orthogonal iteration** is:

1. $AQ_k \to Z$, where k is the iteration and $Q_0 = I$
2. $Z \to Q_{k+1}R_{k+1}$, the QR factorization of Z
3. Repeat $AQ_{k+1} \to Z$ and eventually $Q_k \to Q$

Note in each iteration we are calculating $Q_{k+1}^H A Q_k = R_{k+1}$

6.5.1 Reveal eigenvectors of A from T

Motivation: $A = X\Lambda X^{-1}$ can be hard to calculate.

$$\begin{aligned}
A &= X\Lambda X^{-1} = QR\Lambda R^{-1}Q^H = QTQ^H, \text{ where } T = R\Lambda R^{-1} \\
A &= QY\Lambda Y^{-1}Q^H, \text{ where } T = Y\Lambda Y^{-1} \text{ is easier to compute}
\end{aligned}$$

Focusing on $T = Y\Lambda Y^{-1}$, choose some λ_i (we could get from power or QR iteration).

$$Tx = \lambda_i x \longleftrightarrow (T - \lambda_i I)x = 0 \longleftrightarrow (T - \lambda_i I)x = \begin{bmatrix} T_{11} - \lambda_i I & T_{12} & T_{13} \\ 0 & 0 & T_{23} \\ 0 & 0 & T_{33} - \lambda_i I \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \text{ where one diagonal element is 0}$$

And solve with back substitution:

$X_3 = 0 : (T_{33} - \lambda_i I)X_3 = 0$

X_2 is a free parameter $\in \mathbb{R} : 0X_2 + T_{33}X_3 = 0 \Longrightarrow 0X_2 = 0$

$X_1 = -(T_{11} - \lambda_i I)^{-1}T_{12}X_2 : (T_{11} - \lambda_i I)X_1 + T_{12}X_2 + T_{13}X_3 = 0$

It follows the eigenvectors of A are Qy_i. Note, $(T_{11} - \lambda_i I)$ nonsingular as long as the algebraic multiplicity of λ_i is 1.

6.5.2 Rate of convergence in orthogonal (and QR) iteration

Property: the angle between two subspaces, U and V, is defined as $\left\|UU^T - VV^T\right\|_2$
In orthogonal interation, $span\{q_1, \cdots, q_i\} \longrightarrow X, span\{x_1, \cdots, x_i\}$. Convergence is dictated by how quickly these spans converge. The rate of convergence is $O(|\frac{\lambda_{i+1}}{\lambda_i}|^k)$.

6.6 QR iteration

In the QR iteration, we ask if we can go from T_k to T_{k+1} directly. Observe

$$\begin{aligned} A = Q_k T_k Q_k^H &\Longrightarrow T_k = Q_k^H A Q_k \\ AQ_k = Q_{k+1}R_{k+1} &\Longrightarrow Q_{k+1}^H A = R_{k+1}Q_k^H \\ T_k = Q_k^H(Q_{k+1}R_{k+1}) &\longrightarrow T_k = U_{k+1}R_{k+1} \text{ for } U_{k+1} = Q_k^H Q_{k+1} \\ T_{k+1} = (R_{k+1}Q_k^H)Q_{k+1} &\longrightarrow T_{k+1} = R_{k+1}U_{k+1} \text{ for } U_{k+1} = Q_k^H Q_{k+1} \end{aligned}$$

So we have an algorithm for $T_k \to T_{k+1}$, this process is the **QR iteration**:

1. $T_k \longrightarrow U_{k+1}R_{k+1}$, the QR factorization of T_k
2. $R_{k+1}U_{k+1} \longrightarrow T_{k+1}$
3. Repeat with T_{k+1}

Proof by induction: R_{k+1} is the same in both QR factorization of $A = Q_{k+1}R_{k+1}$ and $T_k = U_{k+1}R_{k+1}$

$$\begin{aligned} &\textit{case } 1: A = AQ_0 = Q_1R_1, A = T_0 = U_1R_1^*, \text{ and } T_1 = Q_k^H AQ_1 \\ &U_1R_1^* = Q_0^T Q_1 R_1 = Q_1 R_1 \implies R_1^* = R_1 \text{ and } U_1 = Q_0^T Q_1 \\ &\textit{case } k: \text{Assume } R_k^* = R_k, U_k = Q_{k-1}^T Q_k, \text{ and } T_k = Q_k^H A Q_k \end{aligned}$$

6.7 QR iteration on upper Hessenberg

Each QR iteration step of a dense matrix is $O(n^3)$. If we run for $O(k)$ iterations, then this algorithm is $O(kn^3)$. To reduce flops, we can first convert A to upper Hessenberg ($H = Q^H AQ$) with $O(n^3)$, and proceed with QR iteration on H using Givens rotations with complexity $O(n^2)$ (so overall complexity is reduced to $O(n^3 + kn^2)$):

$$\text{Choose } Q_1^T = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{P}_1 \end{bmatrix} \text{ to perform a Householder rotation onto the first two entries of } a_1 \in A$$

$$\text{Observe } Q_1^T A Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{P}_1 \end{bmatrix} A \begin{bmatrix} 1 & 0 \\ 0 & \tilde{P_1^T} \end{bmatrix} = \begin{bmatrix} x & x & \cdots & \\ x & x & \cdots & \\ 0 & x & \cdots & \\ \vdots & \vdots & \ddots & \end{bmatrix} \text{ where } a_{11} \text{ is never changed, the rest of } a_1$$

$$\text{is only operated on by } \tilde{P}_1, \text{ and the rest of } a_1^T \text{ is only operated on by } \tilde{P_1^T}$$

$$\text{Continuing on, } Q_n^T \dots Q_2^T Q_1^T A Q_1 Q_2 \dots Q_n = H = Q^H AQ \text{ where } Q_k^T = \begin{bmatrix} I_k & 0 \\ 0 & \tilde{P}_k \end{bmatrix}$$

H **remains upper Hessenberg in QR iteration:** This follows since in the first step of QR iteration, H_k is transformed to R_k with givens rotations, $U_k^H H_k = R_k$. And in the second step of QR iteration, H_{k+1} is created as $R_k U_k = H_{k+1} = U_k^H H_k U_k$. Since U_k is a series of givens rotations, these rotations can be constructed/ordered so that H_{k+1} preserves upper Hessenberg.

6.8 QR iteration with shift

QR iteration with shift accelerates convergence. First observe for $\lambda_i \in \lambda(A) \to (\lambda_i - \mu) \in \lambda(A - \mu I)$. The resulting converence is $|[(\lambda_{i+1} - \mu)/(\lambda_i - \mu)]|^k$. Shift does not require that $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$.

QR iteration with shift process:

1. $\mu_k = T_k[n, n]$
2. $(T_k - \mu_k I) \longrightarrow U_{k+1}R_{k+1}$, QR factorization of the shifted T_k
3. $R_k U_k + \mu_k I \longrightarrow T_{k+1}$, and repeat!

Observe, this shift preserves the original QR iteration:

$$\begin{aligned} (T_k - \mu_I) = U_{k+1}R_{k+1} &\Longrightarrow U_{k+1}^H T_k - \mu_k U_{k+1}^H = R_{k+1} \\ T_{k+1} = R_{k+1}U_{k+1} + \mu_k I &\Longrightarrow T_{k+1} = (U_{k+1}^H T_k - \mu_k U_{k+1}^H)U_{k+1} + \mu_k I \\ T_{k+1} = U_{k+1}^H T_k U_{k+1} - \mu_k I + \mu_k I &= U_{k+1}^H T_k U_{k+1} \end{aligned}$$

6.8.1 Implicit Q theorem

The **implicit Q theorem** tells us that if i) we get any upper Hessneberg, H_{k+1} from a transformation of $H_k \to H_{k+1}$ of the form $U^T H_k U$ ii) $We_1 = Qe_1$ for two such transformations, then the columns of W and Q are equal, up to a sign.

Proof: We show for $A = QHQ^T$, Q orthogonal and H upper Hessenberg, that Q, H are determined by A and Qe_1:

$$\begin{aligned}
AQ &= QH, \text{ assume we know } q_1, \dots, q_k \text{ of } Q \\
A\begin{bmatrix} Q_k & X \end{bmatrix} &= \begin{bmatrix} Q_k & X \end{bmatrix} \begin{bmatrix} H_k & X \\ 0 & X \end{bmatrix}, \ X \text{ unknown and } H_k \in \mathbb{R}^{k \times k} \\
Aq_k &= \sum_{i=1}^{k} h_{i,k} q_i + k_{k+1,k} q_{k+1}, \text{ the kth column of } AQ, \text{ where } q_j^T A q_k = h_{j,k} \\
k_{k+1,k} q_{k+1} &= Aq_k - \sum_{i=1}^{k} h_{i,k} q_i, \text{ the RHS of which is known} \\
\Rightarrow |h_{k+1,k}| &= \left\| Aq_k - \sum_{i=1}^{k} h_{i,k} q_i \right\|_2 \text{ and } q_{k+1} = \frac{Aq_k - \sum_{i=1}^{k} h_{i,k} q_i}{h_{k+1,k}}
\end{aligned}$$

6.8.2 Fracis shift

The **Francis shift** is a way of selecting shifts based on the bottom-right 2×2 block in a way that maintains a real-valued matrix. In effect, we double-shift using complex conjugates, $\mu, \overline{\mu}$:

$$\begin{aligned}
H_{k-1} - \mu I &= U_k R_k \\
H_k &= R_k U_k + \mu I \\
H_k - \overline{\mu} I &= U_{k+1} R_{k+1} \\
H_{k+1} &= R_{k+1} U_{k+1} + \overline{\mu} I \\
H_{k+1} &= U_{k+1}^H H_k U_{k+1} = U_{k+1}^H U_k^H H_{k-1} U_k 1 U_{k+1} = (U_k U_{k+1})^H H_{k-1} (U_k U_{k+1})
\end{aligned}$$

Proof Consider QR factorization to show $(U_1 U_2)$ is real

$$(U_k U_{k+1})(R_{k+1} R_k) = H_{k-1}^2 - (\mu + \overline{\mu}) H_{k-1} + |\mu|^2 I, \text{ where each component of the polynomial is } \in \mathbb{R}$$

From uniqueness of QR factorization, $(U_1 U_2)$ must be real as well. So at any step of the Francis shift, we want $H_{k+1} = Q^T H_{k-1} Q$

6.9 QR iteration with deflation

If any sub-diagonal element of an upper Hessenberg matrix, H, is 0, it can be written as $H = \begin{bmatrix} H_{11} & H_{12} \\ 0 & H_{22} \end{bmatrix}$ with H_{11} and H_{22} upper Hessenberg and $\lambda(H) = \lambda(H_{11}) \cup \lambda(H_{22})$
Theorem: $\lambda(H) = \lambda(H_{11}) \cup \lambda(H_{22})$ for H block upper triangular. **Proof:**

$$\begin{aligned}
&\Longrightarrow Hx = \lambda x \longrightarrow \begin{bmatrix} H_{11} & H_{12} \\ 0 & H_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} H_{11} x_1 + H_{12} x_2 \\ H_{22} x_2 \end{bmatrix} = \begin{bmatrix} \lambda x_1 \\ \lambda x_2 \end{bmatrix} \\
&\quad \text{and either } x_2 = 0 \text{ and } \lambda \in \lambda(H_{11}) \text{ or not and } \lambda \in \lambda(H_{22}) \\
&\Longleftarrow H_{11} p_1 = \lambda p_1 \longrightarrow \begin{bmatrix} H_{11} & H_{12} \\ 0 & H_{22} \end{bmatrix} \begin{bmatrix} p_1 \\ 0 \end{bmatrix} = \begin{bmatrix} H_{11} p_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \lambda p_1 \\ 0 \end{bmatrix} \\
&\Longleftarrow H_{22} p_2 = \lambda p_2 \longrightarrow \begin{bmatrix} H_{11} & H_{12} \\ 0 & H_{22} \end{bmatrix} \begin{bmatrix} x \\ p_2 \end{bmatrix} = \begin{bmatrix} H_{11} x + H_{12} p_2 \\ H_{22} p_2 \end{bmatrix} = \begin{bmatrix} \lambda x_1 \\ 0 \end{bmatrix} \\
&\quad \text{where } H_{11} x + H_{12} p_2 = \lambda x \text{ for } x = -(H_{11} - \lambda I)^{-1} H_{12} p_2, \text{ making } \lambda \in \lambda(H)
\end{aligned}$$

Theorem: If H is singular unreduced upper Hessenberg, then in QR factorization, $H = QR$, the last row of R is zero.
Explanation: When constructing QR iteration, each column of R can be linearly independent from the previous ones (since we're adding a dimension) except for the last one (since H and R must be singluar):

$$h_1 = h_{11} e_1 + h_{21} e_2 \qquad h_2 = h_{12} e_1 + h_{22} e_2 + h_{32} e_3 \qquad h_{n-1} = \sum_{i=1}^{n} h_{n-1,i} e_i$$

6.10 QR iteration on symmetric matrices

Upper Hessenberg symmetric matrices are tri-diagonal matrices

- Unsymmetric case complexity: Transform to upper Hessenberg: $O(n^3)$; QR iteration step: $O(n^2)$; overall QR iteration: $O(pn^3)$, where p is the number of iterations per eval (assume quadratic convergence)

- Symmetric case complexity: Transform to upper Hessenberg: $O(n^3)$; QR iteration step: $O(n)$; overall QR iteration: $O(pn^2)$, where p is the number of iterations per eval (assume cubic convergence)

7 Finding eigenvalues of sparse matrices

7.1 Arnoldi process

The **Arnoldi process** reveals first k eigenvalues of a sparse matrix as follows:

1. Begin with random $q_1 \in Q$, such that $\|q_1\|_2 = 1$

Iterate through each of the first k columns of Q with

2. $Aq_j = \sum_{k=1}^{j+1} h_{kj} q_k$, observing we can recover all h_{ij} for $i \leq j$ since $q_i^T A q_j = h_{ij}$

3. $Aq_j = \sum_{k=1}^{j} h_{kj} q_k + h_{j+1,j} q_{j+1}$

4. $r = Aq_j - \sum_{k=1}^{j} h_{kj} q_k = h_{j+1,j} q_{j+1}$, where only r is unknown

5. $\|q_{j+1}\|_2 = 1 \implies h_{j+1,j} = \|r\|_2$ and $q_{j+1} = \dfrac{r}{h_{j+1,j}}$

Output: k columns of Q and the upper $k \times k$ block of H, which can be used in the QR iteration to reveal k eigenvalues close to $\lambda(A)$:

$$AQ = QH \implies AQ_k = Q_k H_k + h_{k+1,k} q_{k+1} e_k^T, \text{ where } Q_k = Q[:, 1:k], H_k = [1:k, 1:k]$$
$$AQ_k = Q_k X_k \Lambda_k X_k^{-1} + h_{k+1,k} q_{k+1} e_k^T, \text{ where } H_k = X_k \Lambda_k X_k^{-1} \text{ through QR iteration}$$
$$A(Q_k X_k) = (Q_k X_k)\Lambda_k + h_{k+1,k} q_{k+1} x_k^T, \text{ where } x_k^T \text{ is the } k^{th} \text{ column of } X$$

And we get an equation where i) $AQ_k \approx Q_k H_k$, ii) Λ_k contains k eigenvalues close to $\lambda_i \in \lambda(A)$, iii) $(Q_k X_k)$ serve as eigenvectors for those eigenvalues, and iv) $h_{k+1,k} q_{k+1} x_k^T$ represents something like an error term.

7.2 Krylov spaces

A space of sparse Matrix-vector products: $K(A, q, k) = span\{q_1, Aq_1, A^2 q_1, \ldots, A^k q_1\}$

7.2.1 QR factorization of Krylov subspace contains Q_k from Arnoldi

Proof: We show for $K_k = Q_k R_k$, that R_k is upper triangular.

$$\text{Start with } Q^T K_k = R \text{ upper triangular for } K_k = \begin{bmatrix} | & | & & | \\ q_1 & Aq_1 & \ldots & A^k q_1 \\ | & | & & | \end{bmatrix}$$
$$Q^T k_j = Q^T A^{j-1} q_1 = Q^T Q H^{j-1} Q^T q_1, \text{ since } A^k = Q^T H^k Q$$
$$= H^{j-1} Q^T q_1 = H^{j-1} e_1, \text{ since } Q \text{ orthogonal}$$
$$\Rightarrow r_j \in R = h_1 \in H^{j-1}, \text{ which has top } j \text{ rows nonzero}$$

The last statement can be checked by iteratively checking the first column of H^i. This result indicates that $Q_k K_k$, produces an upper right triangular matrix since Q_k is the first k columns of Q. This also means Q_k forms a basis for $K(A, q_1, k)$.

7.2.2 Arnoldi process generates a minimal polynomial

Polynomial properties

- If A is diagonalizable, i.e., $A = X \Lambda X^{-1}$, then polynomial $f(A) = X f(\Lambda) X^{-1}$

- **Characteristic polynomial** of A is $p_A(z) = det(zI - A) = \prod(z - \lambda_i)$ and $p_A(\lambda_i) = 0$ for $\lambda_i \in \lambda(A)$

- $f(A) = 0 \implies \lambda_i \in \lambda(A)$ are the roots of the polynomial (e.g., $p_A(A) = X p_A(\Lambda) X^{-1} = 0$

Our hope is that for $p_k(H_k) = 0$, $p_k(A)$ is minimally small. We show $\|p_K(A)q_1\|_2$ is minimized:

$$\begin{aligned}
f(x) &= x^k + f_{k-1}x^{k-1} + \cdots + f_0, \text{ for } f \text{ that minimizes } \|f(A)q_1\|_2 \\
f(A) &= (A^k + f_{k-1}A^{k-1} + \cdots + f_0)q_1 = A^k q_1 + K_k f, \text{ where } f \text{ is a vector of coefficients} \\
&= A^k q_1 + Q_k y, \text{ for some } y, \text{ since } Q_k \text{ forms a basis for Krylov space} \\
\text{Minimal } \|f(A)q_1\|_2 &\implies \text{minimal} \left\|A^k q_1 + Q_k y\right\|_2, \text{ so we need to choose } y \text{ to minimize polynomial} \\
\text{minimal} \left\|A^k q_1 + Q_k y\right\|_2 &\implies Q_k^T f(A) q_1 = 0 \\
Q_k^T f(A) q_1 &= Q_k^T Q f(A) Q^T q_1 = \begin{bmatrix} I_k & 0 \end{bmatrix} f(H) e_1 = I_k f(H_k) e_1
\end{aligned}$$

This proof shows that $\|f(A)q_1\|_2$ is minimal $\Leftrightarrow I_k f(H_k)e_1 = 0$, which $p_k(H_k)$ achieves since $p_k(H_k) = 0$

7.3 Lanczos process

The **Lanczos process** is a parallel process to the Arnoldi process, but for symmetric matrices. Reminder: A symmetric upper Hessenberg matrix, T is tri-diagonal. The process follows

1. $\alpha_k = q_k^T A q_k \implies \alpha_k q_k = A q_k$
2. $r_k = A q_k - \beta_{k-1} q_{k-1} - \alpha_k q_k \implies r_k = \beta_{k-1} q_{k-1}$, r_k becomes the orthogonal part of $A q_k$
3. $\beta_k = \|r_k\|_2$
4. $q_{k+1} = \dfrac{r_k}{\beta_k}$

The orthogonalization in step 2 is reduced from $O(k)$ in Arnoldi to $O(1)$ in Lanczos because of the symmetry of A

7.3.1 Process for revealing the max eigenvalue of A

$$\begin{aligned}
&\lambda(T_k) \approx \lambda(A) \\
&\lambda_1 \in \lambda(T_k) = \max_{x \neq 0} \frac{y^T Q_k^T A Q y}{\|y\|_2^2}, \text{ by property that } \lambda_1 \in \lambda(A) = \max_{x \neq 0} \frac{x^T A x}{\|x\|_2^2} \\
&\qquad \implies \text{ want max } x \text{ of the form } Q_k y \\
&\qquad \implies \text{ want max } x \text{ in Krylov space, a subspace of } \mathbb{R}^k \\
&\qquad \implies \lambda_1 \in \lambda(T_k) \leq \lambda_1 \in \lambda(A), \text{ since it is the max in a smaller space}
\end{aligned}$$

$$\begin{aligned}
&\lambda_1 \in \lambda(T_k) = \max_{x \neq 0} \frac{q_1^T p(A) A p(A) q_1}{q_1^T p(A)^2 q_1}, \text{ and see textbook for step from here to next step} \\
&\qquad \implies \lambda_1 \in \lambda(T_k) \leq \lambda_1 - (\lambda_1 - \lambda_n)\left(\frac{\tan(\theta)}{T_{k-1}^{Cheb}(1 + 2p_1)}\right), \text{ where } p_1 = \frac{\lambda_1 - \lambda_2}{\lambda_2 - \lambda_n}
\end{aligned}$$

Observe that the RHS approaches λ_1 when λ_1 is well separated from the other eigenvalues.