1 Reduction to Hessenberg and Triangular Form (5.5)

One of the issues with the QR iteration is its computational cost; it's not very efficient. In this section, we'll learn about how we can convert A into a matrix that is faster to compute with.

1.1 Upper Hessenberg Matrix

Definition 1.1: Upper Hessenberg Matrix

An $n \times n$ matrix H is called **upper Hessenberg** if $h_{ij} = 0$ for $i > j + 1$.

Remarks:

- An upper Hessenberg matrix might look like

$$\begin{bmatrix} * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ 0 & * & * & * & * & * & * \\ 0 & 0 & * & * & * & * & * \\ 0 & 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & * & * \end{bmatrix}$$

- If H is also symmetric ($A = A^T$), then we get a tridiagonal matrix. This matrix might look like

$$\begin{bmatrix} * & * & 0 & 0 & 0 & 0 & 0 \\ * & * & * & 0 & 0 & 0 & 0 \\ 0 & * & * & * & 0 & 0 & 0 \\ 0 & 0 & * & * & * & 0 & 0 \\ 0 & 0 & 0 & * & * & * & 0 \\ 0 & 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & * & * \end{bmatrix}$$

(Example.) Is the following matrix

$$H = \begin{bmatrix} 4 & 0 & -1 & 2 \\ -8 & -3 & 5 & 6 \\ 0 & -5 & 2 & 7 \\ 0 & 0 & 4 & 1 \end{bmatrix}$$

an upper Hessenberg matrix?

Yes.

1.2 Revised QR Iteration: Preconditioning

We want to transform A to an upper Hessenberg matrix. This process is known as **preconditioning**. The process is as follows:

1. The idea is that we want to find an H such that

$$A = QHQ^T \text{ or } A = QHQ^*$$

   for some unitary Q matrix. This can be found in finite steps, unlike Schur. Remember that A and H should have the same eigenvalues, since they are similar matrices.

2. Apply QR iteration to H. The QR iteration of H only needs $\mathcal{O}(n^2)$ flops. In addition, the QR iteration on a Hessenberg matrix stays within the set of Hessenberg matrices. That is, for $H_0 = H$ where H is Hessenberg, then all H_k obtained by QR iteration are also Hessenberg matrices.

$$\begin{bmatrix} * & \dots & * \\ * & \dots & * \\ \vdots & & \vdots \\ * & * \dots & * \end{bmatrix} \xrightarrow[\text{Step 1}]{\text{Preconditioning}} \underbrace{\begin{bmatrix} \underline{*} & * & * & \dots & * \\ \underline{*} & \underline{*} & * & \dots & * \\ 0 & \underline{*} & \underline{*} & \dots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \underline{*} \end{bmatrix}}_{\text{H: Hessenberg}} \xrightarrow[\text{Step 2}]{\text{QR Iteration}} \underbrace{\dots}_{H_k} \xrightarrow{\text{QR Iteration}} \underbrace{\begin{bmatrix} * & * & * & \dots & * \\ 0 & * & * & \dots & * \\ 0 & 0 & * & \dots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & * \end{bmatrix}}_{T\text{:Upper Triangular}}$$

The idea is, as we apply the iteration more times, we eventually end up with T; that is,

$$\lim_{k \mapsto \infty} H_k = T.$$

1.2.1 Why Does It Work?

Theorem 1.1

An $A \in \mathbb{R}^{n \times n}$ matrix can be decomposed as $A = QHQ^T$ with H upper Hessenberg and Q orthogonal. For $B \in \mathbb{C}^{n \times n}$, the decomposition becomes $B = UHU^*$ with U being unitary.

Proof. The idea is very similar to QR decomposition in the sense that we need reflectors. Note that the QR process looks like

$$Q_n \dots Q_1 Q_1 A \mapsto R,$$

but with this process the eigenvalues are changed. So, we need

$$Q_n \dots Q_2 Q_1 A Q_1^T Q_2^T \dots Q_n^T \mapsto H.$$

Here, H is similar to A. □

(Example.) Consider

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}.$$

Consider

$$\mathbf{x} = \begin{bmatrix} a_{21} \\ a_{31} \\ a_{41} \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} a_{12} \\ a_{13} \\ a_{14} \end{bmatrix}.$$

This gives us

$$A = \begin{bmatrix} a_{11} & \mathbf{y}^T \\ \mathbf{x} & B \end{bmatrix}.$$

So, we can map $\mathbf{x}$ to $\begin{bmatrix} ||x||_2 \\ 0 \end{bmatrix}$ using a reflector $\hat{Q}_1 \in \mathbb{R}^{3 \times 3}$. Then,

$$Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & \hat{Q}_1 \end{bmatrix} \in \mathbb{R}^{4 \times 4}$$

is still orthogonal. From there,

$$Q_1 A = \begin{bmatrix} 1 & 0 \\ 0 & \hat{Q}_1 \end{bmatrix} \begin{bmatrix} a_{11} & \mathbf{y}^T \\ \mathbf{x} & B \end{bmatrix} = \begin{bmatrix} a_{11} & y^T \\ \hat{Q}_1 \mathbf{x} & \hat{Q}_1 B \end{bmatrix}.$$

From there,

$$Q_1 A Q_1^T = \begin{bmatrix} a_{11} & y^T \\ \hat{Q}_1 \mathbf{x} & \hat{Q}_1 B \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \hat{Q}_1^T \end{bmatrix} = \begin{bmatrix} a_{11} & \mathbf{y}^T \hat{Q}_1^T \\ \hat{Q}_1 \mathbf{x} & \hat{Q}_1 B \hat{Q}_1^T \end{bmatrix} = \begin{bmatrix} a_{11} & * & * & * \\ ||x||_2 & c_{11} & c_{12} & c_{13} \\ 0 & c_{21} & c_{22} & c_{23} \\ 0 & c_{31} & c_{32} & c_{33} \end{bmatrix}.$$

At this point, we now only look at the matrix formed by the c_{ij} elements. We need to do $Q_2(Q_1 A Q_1^T) Q_2^T$ to make c_{31} into 0 using a reflector. Then, we'll get

$$H = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \end{bmatrix}.$$

Remark: If A is $n \times n$, then we need more reflectors, i.e.,

$$Q_{n-1} \dots Q_2 Q_1 A Q_1^T Q_2^T \dots Q_{n-1}^T.$$

1.2.2 QR Iteration on Hessenberg

If H is an Hessenberg matrix, then all H_k obtained via QR iteration are Hessenberg matrices as well. With that said, the QR iteration algorithm is as follows:

1. Find H; $A = QHQ^T$ with H being Hessenberg. We can use the MATLAB command `H = hess(A)` to get the Hessenberg matrix.
2. We can perform QR iteration on H to get $H_0 = H$. Using a for- or while- loop, we can run

   $$[Q_k, R_k] = \mathtt{qr}(H_k) \qquad H_{k+1} = R_k Q_k.$$

   We stop iteration when the subdiagonal of H_{k+1} is smaller than a threshold $\epsilon > 0$. That is, when the underlined portion is less than ϵ.

   $$\begin{bmatrix} * & * & * & * & * & * & * \\ \underline{*} & * & * & * & * & * & * \\ 0 & \underline{*} & * & * & * & * & * \\ 0 & 0 & \underline{*} & * & * & * & * \\ 0 & 0 & 0 & \underline{*} & * & * & * \\ 0 & 0 & 0 & 0 & \underline{*} & * & * \\ 0 & 0 & 0 & 0 & 0 & \underline{*} & * \end{bmatrix}$$

   Put it a different way, when

   $$\sqrt{\sum_{i=1}^{n-1} H_{k+1}(i+1, i)^2} < \epsilon.$$

3. The eigenvalues are on the diagonal of H_{k+1}.

1.2.3 Flop Count

In summary, the flop count is

$$\mathcal{O}\left(\frac{10}{3} n^3\right) + \mathcal{O}\left(n^2 N\right),$$

where N is the number of iterations and n is the size of H. Then,

- $\mathcal{O}\left(\frac{10}{3}n^3\right)$ represents the operation for getting the Hessenberg matrix, and
- $\mathcal{O}\left(n^2 N\right)$ represents the operation of finding the QR decomposition of H.

Note: If we didn't use the Hessenberg matrix, the flop count of QR iteration is $\mathcal{O}(n^3 N)$.
Conclusion: When N is large, Hessenberg will be very helpful in reducing computational costs.

(Example: Hermition Matrices.) The **Hermition Matrix** is a matrix A such that $A = A^*$. If we do an Hessenberg matrix, then[a]

$$A^* = UH^*U^*.$$

In particular,

$$A = A^* \implies UHU^* = UH^*U^* \implies H = H^*.$$

The QR decomposition of a tridiagonal matrix is only $\mathcal{O}(n)$, so the iteration part in the QR iteration reduces to $\mathcal{O}(nN)$. So, less work for

[a] $(CD)^* = D^*C^*$.

Remark: The Hermition matrix mimics the symmetric matrix in the reals.

2 Note on SVD Computation (5.8)

For $A \in \mathbb{R}^{n\times m}, n \geq m$, the eigenvalues of AA^T and A^TA are $\sigma_1^2, \ldots, \sigma_r^2, 0, \ldots, 0$ (with σ_i singular values of A, $r = \text{rank}(A)$). Remember that

- AA^T is an $n \times n$ matrix, and
- A^TA is an $m \times m$ matrix.

So, to find the SVD of A, we can find the eigenvalues and eigenvectors of AA^T and A^TA. Note that both AA^T and A^TA are symmetric.

Then, it follows that

$$\kappa_2(AA^T) = \kappa_2(A)^2.$$

If $\kappa_2(A)$ is large, then for AA^T, the operation is *squared*, thus affecting accuracy. So, we generally want to avoid finding eigenvalues of AA^T and A^TA directly.