Numerical Analysis

10th ed

R L Burden, J D Faires, and A M Burden

Beamer Presentation Slides
Prepared by
Dr. Annette M. Burden
Youngstown State University

September 12, 2015

Chapter 9.1: Linear Algebra and Eigenvalues

Theorem (9.1)

Let A be an $n \times n$ matrix and R_i denote the circle in the complex plane with center a_{ii} and radius $\sum_{j=1,\ j\neq i}^{n} |a_{ij}|$; that is,

$$R_i = \left\{ z \in \mathcal{C} \middle| |z - a_{ii}| \leq \sum_{j=1, j\neq i}^{n} |a_{ij}| \right\},$$

where $\mathcal{C}$ denotes the complex plane. The eigenvalues of A are contained within the union of these circles, $R = \cup_{i=1}^{n} R_i$. Moreover, the union of any k of the circles that do not intersect the remaining $(n-k)$ contains precisely k (counting multiplicities) of the eigenvalues.

Chapter 9.1: Linear Algebra and Eigenvalues

Definition (9.2)

Let $\{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \mathbf{v}^{(3)}, \ldots, \mathbf{v}^{(k)}\}$ be a set of vectors. The set is **linearly independent** if whenever

$$\mathbf{0} = \alpha_1 \mathbf{v}^{(1)} + \alpha_2 \mathbf{v}^{(2)} + \alpha_3 \mathbf{v}^{(3)} + \cdots + \alpha_k \mathbf{v}^{(k)},$$

then $\alpha_i = 0$, for each $i = 0, 1, \ldots, k$. Otherwise the set of vectors is **linearly dependent**.

Theorem (9.3)

Suppose that $\{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \mathbf{v}^{(3)}, \ldots, \mathbf{v}^{(n)}\}$ is a set of n linearly independent vectors in $\mathbb{R}^n$. Then for any vector $\mathbf{x} \in \mathbb{R}^n$ a unique collection of constants $\beta_1, \beta_2, \ldots, \beta_n$ exists with

$$\mathbf{x} = \beta_1 \mathbf{v}^{(1)} + \beta_2 \mathbf{v}^{(2)} + \beta_3 \mathbf{v}^{(3)} + \cdots + \beta_n \mathbf{v}^{(n)}.$$

Chapter 9.1: Linear Algebra and Eigenvalues

Definition (9.4)

Any collection of n linearly independent vectors in $\mathbb{R}^n$ is called a **basis** for $\mathbb{R}^n$.

Theorem (9.5)

If A is a matrix and $\lambda_1, \ldots, \lambda_k$ are distinct eigenvalues of A with associated eigenvectors $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(k)}$, then $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(k)}\}$ is a linearly independent set.

Definition (9.6)

A set of vectors $\{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(n)}\}$ is called **orthogonal** if $(\mathbf{v}^{(i)})^t\mathbf{v}^{(j)} = 0$, for all $i \neq j$. If, in addition, $(\mathbf{v}^{(i)})^t\mathbf{v}^{(i)} = 1$, for all $i = 1, 2, \ldots, n$, then the set is called **orthonormal**.

Theorem (9.7)

An orthogonal set of nonzero vectors is linearly independent.

Chapter 9.1: Linear Algebra and Eigenvalues

Theorem (9.8)

Let $\{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_k}\}$ *be a set of* k *linearly independent vectors in* $\mathbb{R}^n$*. Then* $\{\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_k}\}$ *defined below is a* k *orthogonal set of vectors in* $\mathbb{R}^n$*:*

$$
\begin{aligned}
\mathbf{v_1} &= \mathbf{x_1}, \\
\mathbf{v_2} &= \mathbf{x_2} - \left(\frac{\mathbf{v_1^t x_2}}{\mathbf{v_1^t v_1}}\right)\mathbf{v_1}, \\
\mathbf{v_3} &= \mathbf{x_3} - \left(\frac{\mathbf{v_1^t x_3}}{\mathbf{v_1^t v_1}}\right)\mathbf{v_1} - \left(\frac{\mathbf{v_2^t x_3}}{\mathbf{v_2^t v_2}}\right)\mathbf{v_2}, \\
&\vdots \\
\mathbf{v_k} &= \mathbf{x_k} - \sum_{\mathbf{i}=1}^{\mathbf{k}-1}\left(\frac{\mathbf{v_i^t x_k}}{\mathbf{v_i^t v_i}}\right)\mathbf{v_i}.
\end{aligned}
$$

Chapter 9.2: Orthogonal Matrices and Similarity Transformations

Definition (9.9)

A matrix Q is said to be **orthogonal** if its columns $\{\mathbf{q}_1^t, \mathbf{q}_2^t, \dots, \mathbf{q}_n^t\}$ form an orthonormal set in $\mathbb{R}^n$.

Theorem (9.10)

Suppose that Q is an orthogonal $n \times n$ matrix. Then

(i) *Q is invertible with $Q^{-1} = Q^t$;*

(ii) *For any $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^n$, $(Q\mathbf{x})^{\mathbf{t}}\mathbf{Qy} = \mathbf{x}^{\mathbf{t}}\mathbf{y}$;*

(iii) *For any $\mathbf{x}$ in $\mathbb{R}^n$, $||Q\mathbf{x}||_2 = ||\mathbf{x}||_2$.*

(iv) *Any invertible matrix Q with $Q^{-1} = Q^t$ is orthogonal.*

Definition (9.11)

Two matrices A and B are said to be **similar** if a nonsingular matrix S exists with $A = S^{-1}BS$

Chapter 9.2: Orthogonal Matrices and Similarity Transformations

Theorem (9.12)

Suppose A and B are similar matrices with $A = S^{-1}BS$ and λ is an eigenvalue of A with associated eigenvector $\mathbf{x}$. Then λ is an eigenvalue of B with associated eigenvector $S\mathbf{x}$.

Theorem (9.13)

An $n \times n$ matrix A is similar to a diagonal matrix D if and only if A has n linearly independent eigenvectors. In this case, $D = S^{-1}AS$, where the columns of S consist of the eigenvectors, and the ith diagonal element of D is the eigenvalue of A that corresponds to the ith column of S.

Chapter 9.2: Orthogonal Matrices and Similarity Transformations

Corollary (9.14)

An $n \times n$ matrix A that has n distinct eigenvalues is similar to a diagonal matrix.

Theorem (9.15: Schur's Theorem)

Let A be an arbitrary matrix. A nonsingular matrix U exists with the property that

$$T = U^{-1}AU,$$

where T is an upper-triangular matrix whose diagonal entries consist of the eigenvalues of A.

Chapter 9.2: Orthogonal Matrices and Similarity Transformations

Theorem (9.16)

The $n \times n$ matrix A is symmetric if and only if there exists a diagonal matrix D and an orthogonal matrix Q with $A = QDQ^t$.

Corollary (9.17)

Suppose that A is a symmetric $n \times n$ matrix. There exist n eigenvectors of A that form an orthonormal set, and the eigenvalues of A are real numbers.

Theorem (9.18)

A symmetric matrix A is positive definite if and only if all the eigenvalues of A are positive.

Chapter 9.3: The Power Method

The **Power method** is an iterative technique used to determine the dominant eigenvalue of a matrix—that is, the eigenvalue with the largest magnitude. By modifying the method slightly, it can also used to determine other eigenvalues. One useful feature of the Power method is that it produces not only an eigenvalue, but also an associated eigenvector. In fact, the Power method is often applied to find an eigenvector for an eigenvalue that is determined by some other means.

Chapter 9.3: The Power Method

Algorithm 9.1: POWER METHOD

To approximate the dominant eigenvalue and an associated eigenvector of the $n \times n$ matrix A given a nonzero vector $\mathbf{x}$:

INPUT dimension n; matrix A; vector $\mathbf{x}$; tolerance TOL; maximum number of iterations N.

OUTPUT approximate eigenvalue μ; approximate eigenvector $\mathbf{x}$ (with $||\mathbf{x}||_\infty = 1$) or a message that the maximum number of iterations was exceeded.

```
Step 1 Set $k = 1$.
Step 2 Find the smallest integer $p$ with $1 \leq p \leq n$ and $|x_p| = ||\mathbf{x}||_\infty$.
Step 3 Set $\mathbf{x} = \mathbf{x}/x_p$.
Step 4 While ($k \leq N$) do Steps 5–11.
    Step 5 Set $\mathbf{y} = A\mathbf{x}$.
```

Chapter 9.3: The Power Method

Algorithm 9.1: POWER METHOD

```
    Step 7 Find smallest integer $p$ with $1 \leq p \leq n$ and $|y_p| = |\mathbf{y}||_\infty$.
    Step 8 If $y_p = 0$ then OUTPUT ('Eigenvector', $\mathbf{x}$);
                             OUTPUT ('$A$ has the eigenvalue 0, select a
                                     new vector $\mathbf{x}$ and restart');
                             STOP.
    Step 9 Set $ERR = ||\mathbf{x} - (\mathbf{y}/y_p)||_\infty$;
                   $\mathbf{x} = \mathbf{y}/y_p$.
    Step 10 If $ERR < TOL$ then OUTPUT ($\mu, \mathbf{x}$);
                                   (*The procedure was successful.*)
                                   STOP.
    Step 11 Set $k = k + 1$.
Step 12 OUTPUT ('The maximum number of iterations exceeded');
      (*The procedure was unsuccessful.*)
      STOP.
```

Algorithm 9.1: ACCELERATING CONVERGENCE

The sequence $\{\mu^{(m)}\}$ converges linearly to λ_1, so Aitken's Δ^2 procedure discussed in Section 2.5 can be used to speed the convergence. Implementing the Δ^2 procedure in Algorithm 9.1 is accomplished by modifying the algorithm as follows:

```
Step 1 Set $k = 1$;
            $\mu_0 = 0$;
            $\mu_1 = 0$.
Step 6 Set $\mu = y_p$;
            $\hat{\mu} = \mu_0 - \frac{(\mu_1 - \mu_0)^2}{\mu - 2\mu_1 + \mu_0}$.
Step 10 If $ERR < TOL$ and $k \geq 4$ then OUTPUT $(\hat{\mu}, \mathbf{x})$;   STOP.
Step 11 Set $k = k + 1$;
            $\mu_0 = \mu_1$;
            $\mu_1 = \mu$.
```

Chapter 9.3: The Power Method

Algorithm 9.2: SYMMETRIC POWER METHOD

INPUT dimension n; matrix A; vector $\mathbf{x}$; tolerance TOL; maximum number of iterations N.

OUTPUT approximate eigenvalue μ; approximate eigenvector $\mathbf{x}$ (with $\|\mathbf{x}\|_2 = 1$) or a message that the maximum number of iterations was exceeded.

```
Step 1 Set $k = 1$;
           $\mathbf{x} = \mathbf{x}/\|\mathbf{x}\|_2$.
Step 2 While ($k \leq N$) do Steps 3–8.
    Step 3 Set $\mathbf{y} = A\mathbf{x}$.
    Step 4 Set $\mu = \mathbf{x}^t\mathbf{y}$.
    Step 5 If $\|\mathbf{y}\|_2 = 0$, then OUTPUT ('Eigenvector', $\mathbf{x}$);
                               OUTPUT ('$A$ has eigenvalue 0, select
                                        new vector $\mathbf{x}$ and restart');
                               STOP.
```

Chapter 9.3: The Power Method

Algorithm 9.2: SYMMETRIC POWER METHOD

```
        Step 6 Set $ERR = \left\| \mathbf{x} - \frac{\mathbf{y}}{\|\mathbf{y}\|_2} \right\|_2$;
                    $\mathbf{x} = \mathbf{y}/\|\mathbf{y}\|_2$.
        Step 7 If $ERR < TOL$ then OUTPUT $(\mu, \mathbf{x})$;
                    STOP (*The procedure was successful.*)
        Step 8 Set $k = k + 1$.
Step 9 OUTPUT ('Maximum number of iterations exceeded');
        STOP (*Procedure was unsuccessful.*)
```

Theorem (9.19)

Suppose that A is an $n \times n$ symmetric matrix with eigenvalues λ_1, λ_2, $\ldots$, λ_n. If $\|A\mathbf{x} - \lambda\mathbf{x}\|_2 < \varepsilon$ for some real number λ and vector $\mathbf{x}$ with $\|\mathbf{x}\|_2 = 1$, then

$$\min_{1 \le j \le n} |\lambda_j - \lambda| < \varepsilon.$$

Chapter 9.3: The Power Method

Algorithm 9.3: INVERSE POWER METHOD

To approximate an eigenvalue and an associated eigenvector of the $n \times n$ matrix A given a nonzero vector $\mathbf{x}$:

INPUT dimension n; matrix A; vector $\mathbf{x}$; tolerance TOL; maximum number of iterations N.

OUTPUT approximate eigenvalue μ; approximate eigenvector $\mathbf{x}$ (with $\|\mathbf{x}\|_\infty = 1$) or a message that the maximum number of iterations was exceeded.

```
Step 1 Set $q = \dfrac{\mathbf{x}^t A\mathbf{x}}{\mathbf{x}^t\mathbf{x}}$.
Step 2 Set $k = 1$.
Step 3 Find the smallest integer $p$ with $1 \le p \le n$ and $|x_p| = \|\mathbf{x}\|_\infty$.
Step 4 Set $\mathbf{x} = \mathbf{x}/x_p$.
Step 5 While ($k \le N$) do Steps 6–12.
    Step 6 Solve the linear system $(A - qI)\mathbf{y} = \mathbf{x}$.
```

Algorithm 9.3: INVERSE POWER METHOD

```
    Step 7 If the system does not have a unique solution, then
               OUTPUT (‘$q$ is an eigenvalue’, $q$);
               STOP.
    Step 8 Set $\mu = y_p$.
    Step 9 Find smallest integer $p$ with $1 \le p \le n$ and $|y_p| = \|\mathbf{y}\|_\infty$.
    Step 10 Set $ERR = \|\mathbf{x} - (\mathbf{y}/y_p)\|_\infty$;
                         $\mathbf{x} = \mathbf{y}/y_p$.
    Step 11 If $ERR < TOL$ then set $\mu = (1/\mu) + q$;
                                        OUTPUT $(\mu, \mathbf{x})$;
                                        (The procedure was successful.)
                                        STOP.
    Step 12 Set $k = k + 1$.
Step 13 OUTPUT (‘Maximum number of iterations exceeded’);
      (The procedure was unsuccessful.)
      STOP.
```

Note

Numerous techniques are available for obtaining approximations to the other eigenvalues of a matrix once an approximation to the dominant eigenvalue has been computed. **Deflation techniques** involve forming a new matrix B whose eigenvalues are the same as those of A, except that the dominant eigenvalue of A is replaced by the eigenvalue 0 in B. The following result justifies the procedure.

Chapter 9.3: The Power Method

Theorem (9.20)

Suppose $\lambda_1, \lambda_2, \ldots, \lambda_n$ *are eigenvalues of A with associated eigenvectors* $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(n)}$ *and that* λ_1 *has multiplicity 1. Let* $\mathbf{x}$ *be a vector with* $\mathbf{x}^t\mathbf{v}^{(1)} = 1$. *Then the matrix*

$$B = A - \lambda_1 \mathbf{v}^{(1)}\mathbf{x}^t$$

has eigenvalues $0, \lambda_2, \lambda_3, \ldots, \lambda_n$ *with associated eigenvectors* $\mathbf{v}^{(1)}$, $\mathbf{w}^{(2)}$, $\mathbf{w}^{(3)}, \ldots, \mathbf{w}^{(n)}$, *where* $\mathbf{v}^{(i)}$ *and* $\mathbf{w}^{(i)}$ *are related by the equation*

$$\mathbf{v}^{(i)} = (\lambda_i - \lambda_1)\mathbf{w}^{(i)} + \lambda_1(\mathbf{x}^t\mathbf{w}^{(i)})\mathbf{v}^{(1)},$$

for each $i = 2, 3, \ldots, n$.

Note

There are many choices of the vector **x** that could be used in Theorem 9.20. **Wielandt deflation** proceeds from defining

$$\mathbf{x} = \frac{1}{\lambda_1 v_i^{(1)}}(a_{i1}, a_{i2}, \ldots, a_{in})^t,$$

where $v_i^{(1)}$ is a nonzero coordinate of the eigenvector $\mathbf{v}^{(1)}$, and the values $a_{i1}, a_{i2}, \ldots, a_{in}$ are the entries in the ith row of A.

Algorithm 9.4: WIELANDT DEFLATION

To approximate the second most dominant eigenvalue and an associated eigenvector of the $n \times n$ matrix A given an approximation λ to the dominant eigenvalue, an approximation $\mathbf{v}$ to a corresponding eigenvector, and a vector $\mathbf{x} \in \mathbb{R}^{n-1}$:

INPUT dimension n; matrix A; approximate eigenvalue λ with eigenvector $\mathbf{v} \in \mathbb{R}^n$; vector $\mathbf{x} \in \mathbb{R}^{n-1}$, tolerance TOL, maximum number of iterations N.

OUTPUT approximate eigenvalue μ; approximate eigenvector $\mathbf{u}$ or a message that the method fails.

```
Step 1 Let $i$ be smallest integer with $1 \le i \le n$ and $|v_i| = \max_{1 \le j \le n} |v_j|$.
Step 2 If $i \neq 1$ then
           for $k = 1, \ldots, i-1$
               for $j = 1, \ldots, i-1$
                   set $b_{kj} = a_{kj} - \frac{v_k}{v_i} a_{ij}$.
```

Chapter 9.3: The Power Method

Algorithm 9.4: WIELANDT DEFLATION

```
Step 3 If $i \neq 1$ and $i \neq n$ then
            for $k = i, \ldots, n-1$
                for $j = 1, \ldots, i-1$
                    set $b_{kj} = a_{k+1,j} - \frac{v_{k+1}}{v_i} a_{ij}$;
                        $b_{jk} = a_{j,k+1} - \frac{v_j}{v_i} a_{i,k+1}$.
Step 4 If $i \neq n$ then
            for $k = i, \ldots, n-1$
                for $j = i, \ldots, n-1$
                    set $b_{kj} = a_{k+1,j+1} - \frac{v_{k+1}}{v_i} a_{i,j+1}$.
Step 5 Perform the power method on the $(n-1) \times (n-1)$ matrix
         $B' = (b_{kj})$ with $\mathbf{x}$ as initial approximation.
Step 6 If the method fails, then OUTPUT ('Method fails');   STOP
                                 else let $\mu$ be the approximate eigenvalue &
                                 $\mathbf{w}' = (w'_1, \ldots, w'_{n-1})^t$ the approximate
                                 eigenvector.
```

Chapter 9.3: The Power Method

Algorithm 9.4: WIELANDT DEFLATION

```
Step 7 If $i \neq 1$ then for $k = 1, \ldots, i-1$ set $w_k = w'_k$.
Step 8 Set $w_i = 0$.
Step 9 If $i \neq n$ then for $k = i+1, \ldots, n$ set $w_k = w'_{k-1}$.
Step 10 For $k = 1, \ldots, n$
            set $u_k = (\mu - \lambda)w_k + \left(\sum_{j=1}^{n} a_{ij} w_j\right) \frac{v_k}{v_i}$.
               (Compute the eigenvector using Eq. (9.6).)
Step 11 OUTPUT $(\mu, \mathbf{u})$;   (The procedure was successful.)
        STOP.
```

Chapter 9.4: Householder's Method

Definition (9.21)

Let $\mathbf{w} \in \mathbb{R}^n$ with $\mathbf{w}^t\mathbf{w} = 1$. The $n \times n$ matrix

$$P = I - 2\mathbf{w}\mathbf{w}^t$$

is called a **Householder transformation**.

Theorem (9.22)

A Householder transformation, $P = I - 2\mathbf{w}\mathbf{w}^t$, is symmetric and orthogonal, so $P^{-1} = P$.

Chapter 9.4: Householder's Method

Algorithm 9.5 HOUSEHOLDER'S METHOD

To obtain a symmetric tridiagonal matrix $A^{(n-1)}$ similar to the symmetric matrix $A = A^{(1)}$, construct the following matrices $A^{(2)}, A^{(3)}, \ldots, A^{(n-1)}$, where $A^{(k)} = (a_{ij}^{(k)})$ for each $k = 1, 2, \ldots, n-1$:

```
INPUT dimension $n$; matrix $A$.
OUTPUT $A^{(n-1)}$.    (At each step, A can be overwritten.)
Step 1 For $k = 1, 2, \ldots, n-2$ do Steps 2–14.
    Step 2 Set
        $$q = \sum_{j=k+1}^{n} \left(a_{jk}^{(k)}\right)^2.$$
    Step 3 If $a_{k+1,k}^{(k)} = 0$ then set $\alpha = -q^{1/2}$
        else set $\alpha = -\frac{q^{1/2} a_{k+1,k}^{(k)}}{|a_{k+1,k}^{(k)}|}$.
```

Algorithm 9.5 HOUSEHOLDER’S METHOD

```
Step 4 Set $RSQ = \alpha^2 - \alpha a^{(k)}_{k+1,k}$.    $(Note: RSQ = 2r^2)$
Step 5 Set $v_k = 0$;    $(Note: v_1 = \cdots = v_{k-1} = 0; \textit{not needed.})$
           $v_{k+1} = a^{(k)}_{k+1,k} - \alpha$;
       For $j = k+2, \ldots, n$ set $v_j = a^{(k)}_{jk}$.
       $\left(Note: \mathbf{w} = \left(\frac{1}{\sqrt{2RSQ}}\right)\mathbf{v} = \frac{1}{2r}\mathbf{v}.\right)$
Step 6 For $j = k, k+1, \ldots, n$ set $u_j = \left(\frac{1}{RSQ}\right)\sum_{i=k+1}^{n} a^{(k)}_{ji} v_i$.
       $\left(Note: \mathbf{u} = \left(\frac{1}{RSQ}\right)A^{(k)}\mathbf{v} = \frac{1}{2r^2}A^{(k)}\mathbf{v} = \frac{1}{r}A^{(k)}\mathbf{w}.\right)$
```

Algorithm 9.5 HOUSEHOLDER’S METHOD

Step 7 Set $PROD = \sum_{i=k+1}^{n} v_i u_i$.

$$\left(Note\colon\ PROD = \mathbf{v}^t\mathbf{u} = \frac{1}{2r^2}\mathbf{v}^t A^{(k)}\mathbf{v}. \right)$$

Step 8 For $j = k, k+1, \ldots, n$ set $z_j = u_j - \left(\frac{PROD}{2RSQ}\right) v_j$.

$$\left(Note\colon\ \mathbf{z} = \mathbf{u} - \frac{1}{2RSQ}\mathbf{v}^t\mathbf{u}\mathbf{v} = \mathbf{u} - \frac{1}{4r^2}\mathbf{v}^t\mathbf{u}\mathbf{v} \right.$$

$$\left. = \mathbf{u} - \mathbf{w}\mathbf{w}^t\mathbf{u} = \frac{1}{r}A^{(k)}\mathbf{w} - \mathbf{w}\mathbf{w}^t\frac{1}{r}A^{(k)}\mathbf{w}. \right)$$

Chapter 9.4: Householder's Method

Algorithm 9.5 HOUSEHOLDER'S METHOD

```
    Step 9 For $l = k+1, k+2, \ldots, n-1$ do Steps 10 and 11.
    (Note: Compute $A^{(k+1)} = A^{(k)} - \mathbf{vz}^t - \mathbf{zv}^t = (I - 2\mathbf{ww}^t)A^{(k)}(I - 2\mathbf{ww}^t)$.)
        Step 10 For $j = l+1, \ldots, n$ set
                $a_{jl}^{(k+1)} = a_{jl}^{(k)} - v_l z_j - v_j z_l;$
                $a_{lj}^{(k+1)} = a_{jl}^{(k+1)}.$
        Step 11 Set $a_{ll}^{(k+1)} = a_{ll}^{(k)} - 2v_l z_l.$
    Step 12 Set $a_{nn}^{(k+1)} = a_{nn}^{(k)} - 2v_n z_n.$
    Step 13 For $j = k+2, \ldots, n$ set $a_{kj}^{(k+1)} = a_{jk}^{(k+1)} = 0.$
    Step 14 Set $a_{k+1,k}^{(k+1)} = a_{k+1,k}^{(k)} - v_{k+1} z_k;$
            $a_{k,k+1}^{(k+1)} = a_{k+1,k}^{(k+1)}.$
                (Note: The other elements of $A^{(k+1)}$ are the same as $A^{(k)}$.)
Step 15 OUTPUT $(A^{(n-1)})$; STOP
        (Process complete. $A^{(n-1)}$ is symmetric, tridiagonal, & similar to $A$.)
```

Chapter 9.5: The QR Algorithm

Definition

A **rotation matrix** P differs from the identity matrix in at most four elements. These four elements are of the form

$$p_{ii} = p_{jj} = \cos\theta \quad \text{and} \quad p_{ij} = -p_{ji} = \sin\theta,$$

for some θ and some $i \neq j$.

Chapter 9.5: The QR Algorithm

Algorithm 9.6 QR METHOD

To obtain the eigenvalues of the symmetric, tridiagonal $n \times n$ matrix

$$A \equiv A_1 = \begin{bmatrix} a_1^{(1)} & b_2^{(1)} & 0 & & 0 \\ b_2^{(1)} & a_2^{(1)} & & & \\ 0 & & & & 0 \\ & & & & b_n^{(1)} \\ 0 & & 0 & b_n^{(1)} & a_n^{(1)} \end{bmatrix}$$

INPUT n; $a_1^{(1)}, \ldots, a_n^{(1)}, b_2^{(1)}, \ldots, b_n^{(1)}$; tolerance TOL; maximum number of iterations M.

OUTPUT eigenvalues of A, or recommended splitting of A, or a message that the maximum number of iterations was exceeded.

Chapter 9.5: The QR Algorithm

Algorithm 9.6 QR METHOD

```
Step 1 Set $k = 1$; $SHIFT = 0$.   (Accumulated shift.)
Step 2 While $k \leq M$ do Steps 3–19.
          (Steps 3–7 test for success.)
          Step 3 If $|b_n^{(k)}| \leq TOL$ then set $\lambda = a_n^{(k)} + SHIFT$;
                                             OUTPUT ($\lambda$);
                                             set $n = n - 1$.
          Step 4 If $|b_2^{(k)}| \leq TOL$ then set $\lambda = a_1^{(k)} + SHIFT$;
                                             OUTPUT ($\lambda$);
                                             set $n = n - 1$;
                                                 $a_1^{(k)} = a_2^{(k)}$;
                                             for $j = 2, \ldots, n$
                                                 set $a_j^{(k)} = a_{j+1}^{(k)}$;
                                                     $b_j^{(k)} = b_{j+1}^{(k)}$.
          Step 5 If $n = 0$ then STOP.
```

Chapter 9.5: The QR Algorithm

Algorithm 9.6 QR METHOD

```
Step 6 If $n = 1$ then
            set $\lambda = a_1^{(k)} + SHIFT$;
            OUTPUT ($\lambda$);
            STOP.
Step 7 For $j = 3, \ldots, n-1$
if $|b_j^{(k)}| \leq TOL$ then
OUTPUT ('split into', $a_1^{(k)}, \ldots, a_{j-1}^{(k)}, b_2^{(k)}, \ldots, b_{j-1}^{(k)}$,
'and',
$a_j^{(k)}, \ldots, a_n^{(k)}, b_{j+1}^{(k)}, \ldots, b_n^{(k)}, SHIFT$);
            STOP.
Step 8 (Compute shift.)
      Set $b = -(a_{n-1}^{(k)} + a_n^{(k)}); \quad c = a_n^{(k)} a_{n-1}^{(k)} - \left[b_n^{(k)}\right]^2$;
      $d = (b^2 - 4c)^{1/2}$.
```

Chapter 9.5: The QR Algorithm

Algorithm 9.6 QR METHOD

```
Step 9 If $b > 0$ then set $\mu_1 = -2c/(b+d)$;
                           $\mu_2 = -(b+d)/2$
                  else set $\mu_1 = (d-b)/2$;
                           $\mu_2 = 2c/(d-b)$.
Step 10 If $n = 2$ then set $\lambda_1 = \mu_1 + SHIFT$;
                            $\lambda_2 = \mu_2 + SHIFT$;
                            OUTPUT $(\lambda_1, \lambda_2)$;
                            STOP.
Step 11 Choose $\sigma$ so $|\sigma - a_n^{(k)}| = \min\{|\mu_1 - a_n^{(k)}|, |\mu_2 - a_n^{(k)}|\}$.
Step 12 (Accumulate the shift.)
        Set $SHIFT = SHIFT + \sigma$.
Step 13 (Perform shift.)
        For $j = 1, \ldots, n$, set $d_j = a_j^{(k)} - \sigma$.
```

Chapter 9.5: The QR Algorithm

Algorithm 9.6 QR METHOD

```
Step 14 (*Steps 14 and 15 compute $R^{(k)}$.*)
            Set $x_1 = d_1$;  $y_1 = b_2$.
Step 15 For $j = 2, \ldots, n$
                set $z_{j-1} = \left\{ x_{j-1}^2 + \left[ b_j^{(k)} \right]^2 \right\}^{1/2}$;
                    $c_j = \dfrac{x_{j-1}}{z_{j-1}}$;  $\sigma_j = \dfrac{b_j^{(k)}}{z_{j-1}}$;
                    $q_{j-1} = c_j y_{j-1} + s_j d_j$;
                    $x_j = -\sigma_j y_{j-1} + c_j d_j$;
                If $j \neq n$ then set $r_{j-1} = \sigma_j b_{j+1}^{(k)}$;  $y_j = c_j b_{j+1}^{(k)}$.
                $\left( A_j^{(k)} = P_j A_{j-1}^{(k)}\right.$ *just computed &* $\left. R^{(k)} = A_n^{(k)}. \right)$
```

Chapter 9.5: The QR Algorithm

Algorithm 9.6 QR METHOD

```
        Step 16 (*Steps* 16–18 *compute* $A^{(k+1}$.)
                Set $z_n = x_n$;
                    $a_1^{(k+1)} = \sigma_2 q_1 + c_2 z_1$;
                    $b_2^{(k+1)} = \sigma_2 z_2$.
        Step 17 For $j = 2, 3, \ldots, n-1$
                set $a_j^{(k+1)} = \sigma_{j+1} q_j + c_j c_{j+1} z_j$;
                    $b_{j+1}^{(k+1)} = \sigma_{j+1} z_{j+1}$.
        Step 18 Set $a_n^{(k+1)} = c_n z_n$.
        Step 19 Set $k = k + 1$.
Step 20 OUTPUT ('Maximum number of iterations exceeded');
    (*The procedure was unsuccessful*.)
    STOP.
```

Chapter 9.6: Singular Value Decomposition

Definition (9.24)

Let A be an $m \times n$ matrix.

(i) The **Rank** of A, denoted $RankA$ is the number of linearly independent rows in A.

(ii) The **Nullity** of A, denoted Nullity(A), is $n -$ Rank (A), and describes the largest set of linearly independent vectors $\mathbf{v}$ in $\mathbb{R}^n$ for which $A\mathbf{v} = \mathbf{0}$.

Theorem (9.25)

The number of linearly independent rows of an $m \times n$ matrix A is the same as the number of linearly independent columns of A.

Chapter 9.6: Singular Value Decomposition

Theorem (9.26)

Let A be $m \times n$ matrix.

(i) *The matrices A^tA and AA^t are symmetric.*

(ii) *Nullity $(A) =$ Nullity (A^tA).*

(iii) *Rank $(A) =$ Rank (A^tA).*

(iv) *The eigenvalues of A^tA are real and nonnegative.*

(v) *The nonzero eigenvalues of AA^t are the same as the nonzero eigenvalues of A^tA .*

Definition (9.27)

The **singular values** of an $m \times n$ matrix A are the positive square roots of the nonzero eigenvalues of the $n \times n$ symmetric matrix A^tA. The matrix S is an $m \times n$ matrix with the singular values of A on its diagonal in decreasing order and zeros elsewhere.

Constructing V in the factorization $A = U\,S\,V^t$

The $n \times n$ matrix A^tA is symmetric, so by Theorem 9.16 in Section 9.2, we have a factorization

$$A^tA = V\,D\,V^t,$$

where D is a diagonal matrix whose diagonal consists of the eigenvalues of A^tA, and V is an orthogonal matrix whose ith column is an eigenvector with l_2 norm 1 corresponding to the eigenvalue on the ith diagonal of D.

The specific diagonal matrix depends on the order of the eigenvalues along the diagonal.

Choose D so that these are written in decreasing order. The columns, denoted $\mathbf{v_1^t}, \mathbf{v_2^t}, \ldots, \mathbf{v_n^t}$, of the $n \times n$ orthogonal matrix V are orthonormal eigenvectors corresponding to these eigenvalues.

Constructing U in the factorization $A = U\,S\,V^t$

To construct the $m \times m$ matrix U, we consider the nonzero values $s_1 \geq s_2 \geq \cdots \geq s_k > 0$ and the corresponding columns in V given by $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_k}$. We define

$$\mathbf{u_i} = \frac{\mathbf{1}}{\mathbf{s_i}}\mathbf{A}\mathbf{v_i}, \quad \text{for } i = 1, 2, \ldots, k.$$

Use these as the first k of the m columns of U. These k columns of U form an orthonormal set of vectors in $\mathbb{R}^m$. However, we need $m - k$ additional columns of U.
For this we first need to find $m - k$ vectors which when added to the vectors from the first k columns will give us a linearly independent set. Then we can apply the Gram-Schmidt process to obtain appropriate additional columns.
The matrix U will not be unique unless $k = m$, and then only if all the eigenvalues of $A^t A$ are unique.

An alternative method for finding $A = U\,S\,V^t$

To determine the Singular Value Decomposition of the $m \times n$ matrix A we can:

- Find the eigenvalues $s_1^2 \geq s_2^2 \geq \cdots \geq s_k^2 \geq s_{k+1} = \cdots = s_n = 0$ for the symmetric matrix A^tA, and place the positive square root of s_i^2 in the entry $(S)_{ii}$ of the $m \times n$ matrix S.
- Find a set of orthonormal eigenvectors $\{\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}\}$ corresponding to the eigenvalues of A^tA and construct the $n \times n$ matrix V with these vectors as columns.
- Form $\{\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_k}\}$ as before. Then add a set of orthonormal eigenvectors corresponding to the zero eigenvalues of AA^t and construct the $m \times m$ matrix U with these vectors as columns.

Then A has the Singular Value Decomposition $A = U\,S\,V^t$.

Chapter 9.6: Singular Value Decomposition

LEAST SQUARES APPROXIMATION

Singular value decomposition has application in many areas. One is an alternative for finding the least squares polynomials for fitting data.

Let A be an $m \times n$ matrix, with $m > n$, and $\mathbf{b}$ a vector in $\mathbb{R}^m$. The least squares objective is to find a vector $\mathbf{x}$ in $\mathbb{R}^n$ that will minimize $||A\mathbf{x} - \mathbf{b}||_2$.

Suppose that the singular value decomposition of A is known, that is

$$A = U\,S\,V^t,$$

where U is an $m \times m$ orthogonal matrix, V is an $n \times n$ orthogonal matrix, and S is an $m \times n$ matrix that contains the nonzero singular values in decreasing order along the main diagonal in the first $k \leq n$ rows, and zero entries elsewhere.

LEAST SQUARES APPROXIMATION

Because U and V are both orthogonal we have $U^{-1} = U^t$, $V^{-1} = V^t$, and by part(iii) of Theorem 9.10 in Section 9.2, U and V are both l_2-norm preserving. As a consequence,

$$||A\mathbf{x} - \mathbf{b}||_2 = ||\mathbf{U}\,\mathbf{S}\,\mathbf{V^t x} - \mathbf{U}\,\mathbf{U^t b}||_2 = ||\mathbf{S}\,\mathbf{V^t x} - \mathbf{U^t b}||_2.$$

Let $\mathbf{z} = \mathbf{V^t x}$ and $\mathbf{c} = \mathbf{U^t b}$. Then

$$\begin{aligned} ||A\mathbf{x} - \mathbf{b}||_2 =& ||(s_1 z_1 - c_1, s_2 z_2 - c_2, \ldots, s_k z_k - c_k, -c_{k+1}, \ldots, -c_m)^t||_2 \\ &= \left\{ \sum_{i=1}^{k} (s_i z_i - c_i)^2 + \sum_{i=k+1}^{m} (c_i)^2 \right\}^{1/2}. \end{aligned}$$

Chapter 9.6: Singular Value Decomposition

LEAST SQUARES APPROXIMATION

The norm is minimized when the vector **z** is chosen with

$$z_i = \begin{cases} \dfrac{c_i}{s_i}, & \text{when } i \leq k, \\ \text{arbitrarily}, & \text{when } k < i \leq n. \end{cases}$$

Because $\mathbf{c} = U^t\mathbf{b}$ and $\mathbf{x} = V\mathbf{z}$ are both easy to compute, the least squares solution is also easily found.

Chapter 9.6: Singular Value Decomposition

SVD important uses

1. permits us to obtain the most important features of an $m \times n$ matrix using a matrix that is often of significantly smaller size
2. helps us in determine effective condition numbers for square matrices
3. helps us in determining the effective rank of a matrix
4. helps in removing signal noise.