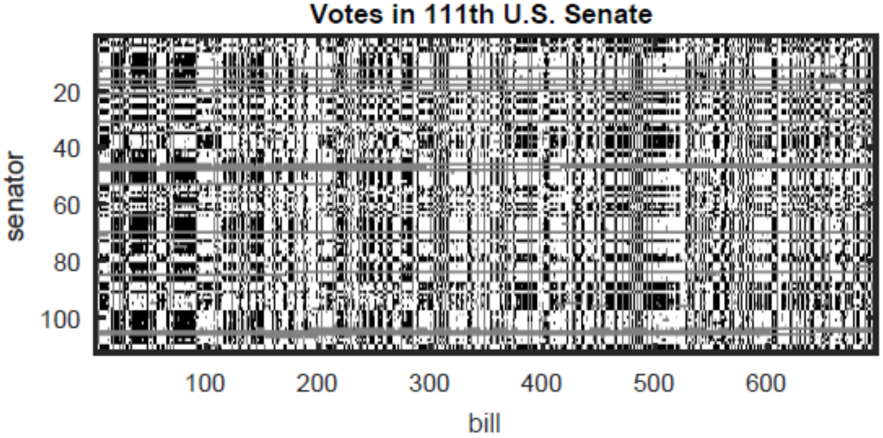
Chapter 7
Matrix Analysis



Section 7.1
From matrix to insight

From matrix to insight

- Any two-dimensional array of numbers may be interpreted as a matrix
- These may come from widely disparate tasks
- Examples:
  - Text/document search
  - Voting patterns
  - Preferences/ratings
  - Graphs
  - Networks: social, political, co-authorship, casting in movies,…
  - Images

Examples of matrices

- A *term-document matrix* may be used for analyzing a body of documents (or *corpus*)
- Each column may be a document; each row a term
- E.g, your textbook may have words like “numerical,” “discretization,” “matrix,” “integration” and “function”
- An analysis textbook may have words like “integration,” “function,” “continuous” and so forth
- The occurrence of “function” may be often in both books, but the other terms are likely to be much different in frequency
- Meaning could be inferred from this kind of approach: latent semantic analysis

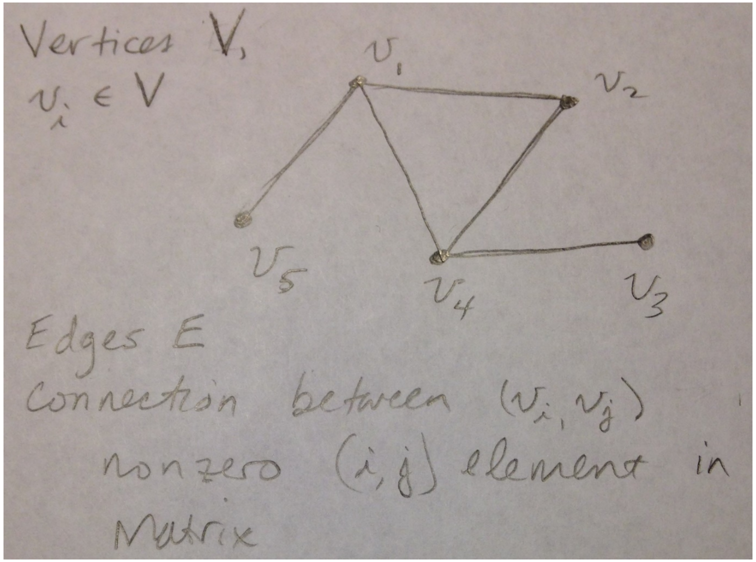
Examples of matrices

- A *term-document matrix* example: matrix in green

Term	FNC by TAD+RJB	Analysis by Rudin	SM by Trefethen	NYT coffee table book
Numerical	251	2	179	0
Integration	37	275	33	18
Function	175	345	123	0
Matrix	151	11	87	0
Continuous	15	212	11	0
Spectral	15	0	124	0
Citizen	0	0	0	13

Graphs as matrices

- Could be social network, internet (or subset of it), the web, etc
- Consider graphs first
- A graph is a set of nodes V connected by set of edges E
- Sometimes the graph is denoted G(V,E)

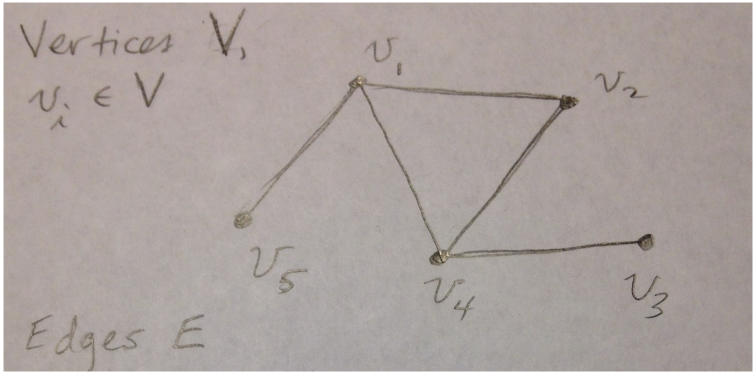


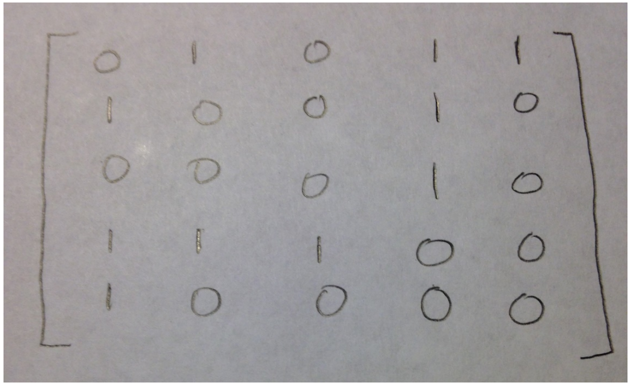
Vertices V, $v_i \in V$

Edges E
connection between (v_i, v_j)
nonzero (i,j) element in Matrix

Adjacency matrix

- We want adjacency matrix $A = \{a_{ij}\}$ that represents this graph
- Edges are unweighted and undirected
- If an edge between (v_i, v_j), then a one is placed in both a_{ij} and a_{ji} (symmetric)
- No self-connections
- Corresponding matrix at right

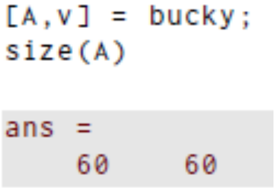




$$\begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

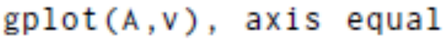
Adjacency matrix: Buckyball

- Matlab has a built-in example of a graph representing the arrangements of carbon atoms in a C_{60} molecule, a.k.a. the buckyball:

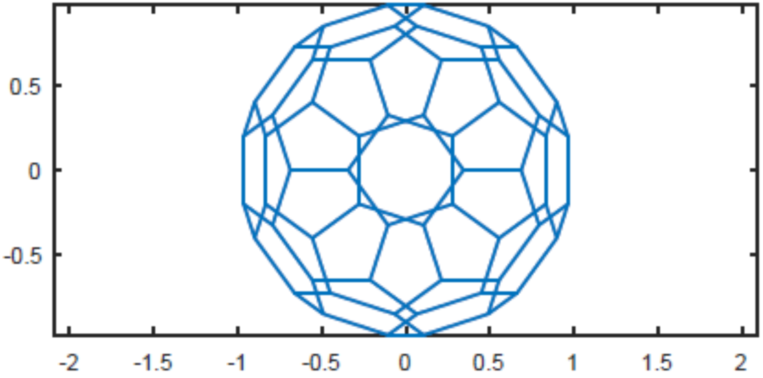


```
[A,v] = bucky;
size(A)

ans =
     60     60
```

- The output has the adjacency matrix A and the vertex locations v
- Plotting the graph:

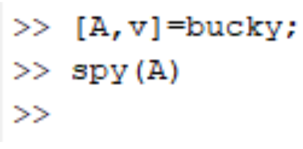


```
gplot(A,v), axis equal
```



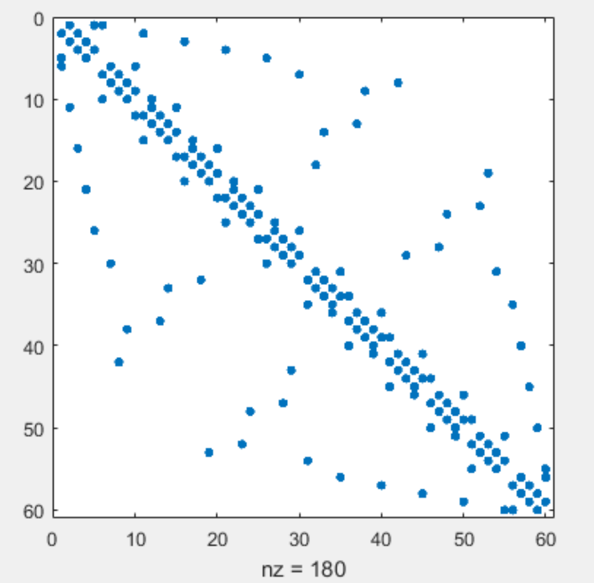
Adjacency matrix: Buckyball

- We know there are 60 nodes; how many edges?
- There is more than one way to compute this
- For undirected, you could use the spy command:



```
>> [A,v]=bucky;
>> spy(A)
>>
```

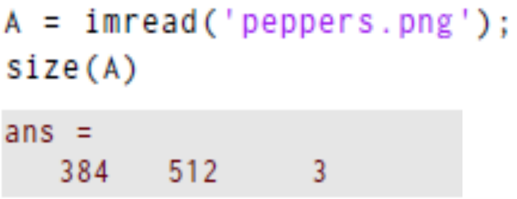
- Is nnz it? Not quite
- You could use the `triu` and `sum` commands; how?



[Example 7.1.4]

Images

- Pictures and images are matrices in matlab
- Use `imread` and `imshow` to display them



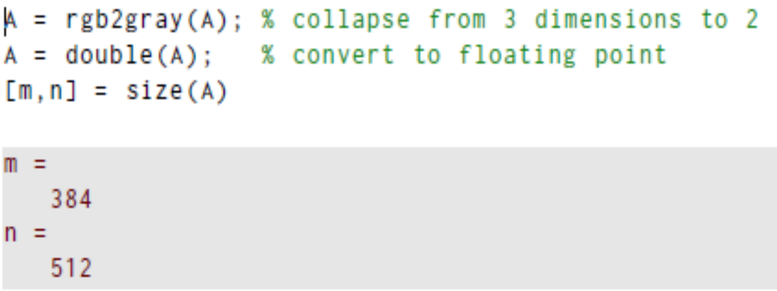
```
A = imread('peppers.png');
size(A)
```

```
ans =
   384   512     3
```

- Three “layers” are RGB components
- To display: `imshow(A)`

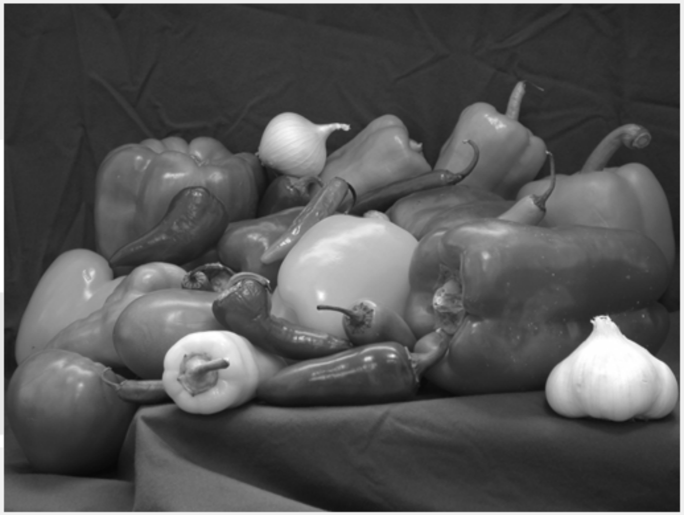
Images

- We can convert to grayscale to get only a 2D matrix
- Use `rgb2gray` and `double`:



```
A = rgb2gray(A); % collapse from 3 dimensions to 2
A = double(A);   % convert to floating point
[m,n] = size(A)
```

```
m =
   384
n =
   512
```



- To display: `imshow(A,[0,255])`
- What if different range spec'd?

[Example 7.1.5]

Section 7.2
Eigenvalue decomposition

The eigenvalue decomposition

- We can decompose a matrix $\boldsymbol{A}$, under some assumptions, into a useful set of other matrices using its eigenvalues and eigenvectors.
- Recall that the eigenvalue problem is $\boldsymbol{Ax} = \lambda\boldsymbol{x}$, for the eigenvalues λ and eigenvectors $\boldsymbol{x}$
- (Sometimes the eigenvectors are referred to as the eigenspaces.)
- We need to review some results and terminology from linear algebra before proceeding
- In particular we need to recall the complex-valued case

Complex vectors and matrices

- Recall complex numbers $x = a + ib$, where $i^2 = -1$ and a, b are real-valued.
- The complex conjugate is $\bar{x} = a - ib$
- For a matrix $\boldsymbol{A}$ with complex-valued elements, the transpose needs to use complex conjugation at the same time: $\boldsymbol{A}^* = (\overline{\boldsymbol{A}})^T = \overline{\boldsymbol{A}^T}$
- The *hermitian* will keep certain properties analogous with the real-valued case
- e.g., $(\boldsymbol{A}^T\boldsymbol{A})^T = \boldsymbol{A}^T\boldsymbol{A}$ so that $\boldsymbol{A}^T\boldsymbol{A}$ is symmetric. So is $(\boldsymbol{A}^*\boldsymbol{A})^* = \boldsymbol{A}^*\boldsymbol{A}$
- Essentially, hermitian replaces transpose for complex-valued matrices and vectors

Complex vectors and matrices

- Essentially, hermitian replaces transpose for complex-valued matrices and vectors
- For inner products with complex $\boldsymbol{u}$ and $\boldsymbol{v}$: $\boldsymbol{u}^*\boldsymbol{v} = \sum_{k=1}^{n} \overline{u}_k v_k,$
- This defines the two-norm for vectors, $\left|\left|\boldsymbol{u}\right|\right|_2 = \sqrt{\boldsymbol{u}^*\boldsymbol{u}}$, and this will in turn define the two norm for matrices
- Definition of orthogonal
- And orthonormal set of vectors $\boldsymbol{u}_j, j = 1,2, \dots, n$: $\boldsymbol{u}_i^*\boldsymbol{u}_j = 1$ if $i = j$, otherwise $\boldsymbol{u}_i^*\boldsymbol{u}_j = 0$

Complex vectors and matrices

- A square real-valued matrix with orthonormal columns is an orthogonal matrix
- A square complex matrix with orthonormal columns is *unitary*
- An $n \times n$ unitary matrix $\boldsymbol{U}$ satisfies $\boldsymbol{U}^{-\mathbf{1}} = \boldsymbol{U}^*$and $\left||\boldsymbol{U}\boldsymbol{x}|\right|_2 = \left||\boldsymbol{x}|\right|_2$ for any complex vector $\boldsymbol{x} \in \mathbb{C}^n$

The eigenvalue problem

- The eigenvalue problem is $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$, for the eigenvalues λ and eigenvectors x
- It can also be written $0 = (\lambda\boldsymbol{I} - \boldsymbol{A})\boldsymbol{x}$
- We find eigenvalues λ_k such that $\lambda_k\boldsymbol{I} - \boldsymbol{A}$ is singular, and we find the corresponding eigenvectors $\boldsymbol{x}_k$ for $k = 1,2, \dots, n$
- When doing this by hand, one calculates the roots of the characteristic polynomial $\det(\lambda\boldsymbol{I} - \boldsymbol{A})$
- There are thus n eigenvalues for an $n{\times}n$ matrix, counting multiplicity
- Note: eigenvalues and eigenvectors not done that way in computer

The eigenvalue decomposition

- Suppose we know the eigenvalues λ_k and eigenvectors $\boldsymbol{v}_k$ so that $\boldsymbol{A}\boldsymbol{v}_k = \lambda_k \boldsymbol{v}_k$, for $k = 1,2,\ldots,n$
- For each k, we have a column vector on each side of the equation
- We can assemble each of those n column vectors into a matrix:

$$\rightarrow \quad \begin{bmatrix} \boldsymbol{A}\boldsymbol{v}_1 & \boldsymbol{A}\boldsymbol{v}_2 & \cdots & \boldsymbol{A}\boldsymbol{v}_n \end{bmatrix} = \begin{bmatrix} \lambda_1 \boldsymbol{v}_1 & \lambda_2 \boldsymbol{v}_2 & \cdots & \lambda_n \boldsymbol{v}_n \end{bmatrix}$$

- Rewrite: →

$$\boldsymbol{A}\begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}$$

- Matrix form: →

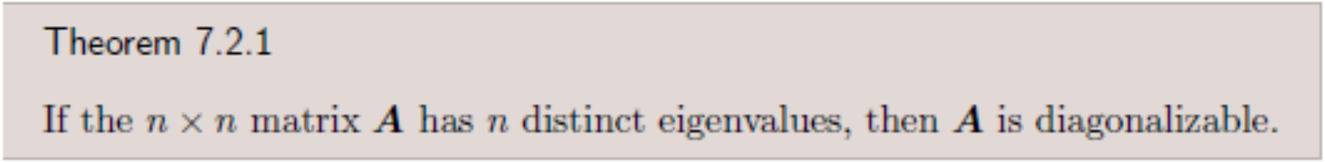
$$\boldsymbol{A}\boldsymbol{V} = \boldsymbol{V}\boldsymbol{D}. \tag{7.2.3}$$

The eigenvalue decomposition

- If we know that there is a complete set of linearly independent (LI) eigenvectors $\boldsymbol{v}_k$, then $\boldsymbol{V}^{-1}$ exists, and we can postmultiply by it to get:

$$\boldsymbol{A} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^{-1}$$

- This is the eigenvalue decomposition, or EVD, of $\boldsymbol{A}$
- If $\boldsymbol{A}$ is diagonalizable, it will have an EVD
- When does that happen?

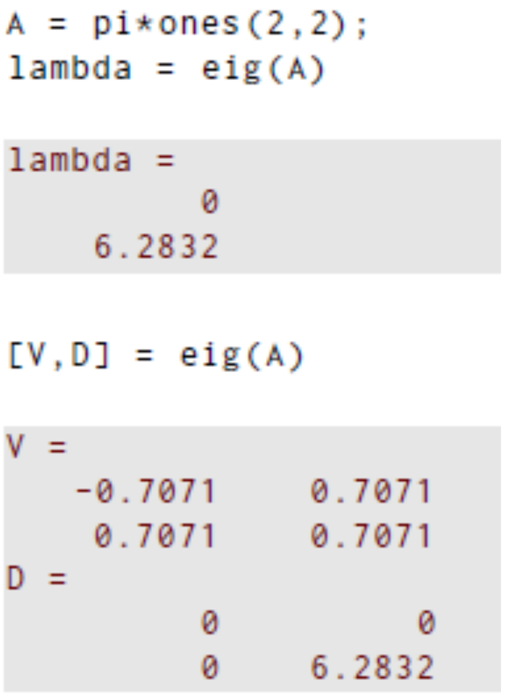


Theorem 7.2.1

If the $n \times n$ matrix $\boldsymbol{A}$ has n distinct eigenvalues, then $\boldsymbol{A}$ is diagonalizable.

Computing EVD in Matlab

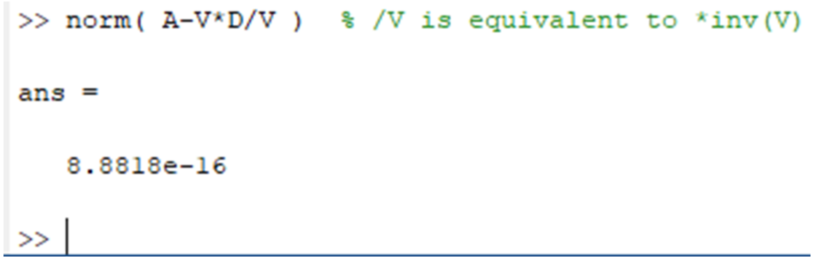
- We can use the `eig` command to compute the EVD
- The eigenvalues arrive as the diagonal elements of $\boldsymbol{D}$
- The eigenvectors are the columns of $\boldsymbol{V}$
- $\boldsymbol{A}$ is singular, but has distinct eigenvalues, and because of that, it has a complete set of LI eigenvectors. Because of that, $\boldsymbol{V}$ is nonsingular



```
A = pi*ones(2,2);
lambda = eig(A)

lambda =
         0
    6.2832

[V,D] = eig(A)

V =
   -0.7071    0.7071
    0.7071    0.7071
D =
         0         0
         0    6.2832
```

Computing EVD in Matlab

- We can easily check that the EVD is equivalent to A:

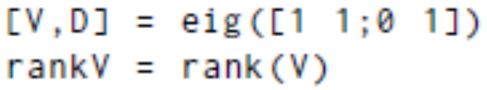


```
>> norm( A-V*D/V )  % /V is equivalent to *inv(V)

ans =

   8.8818e-16

>> 
```

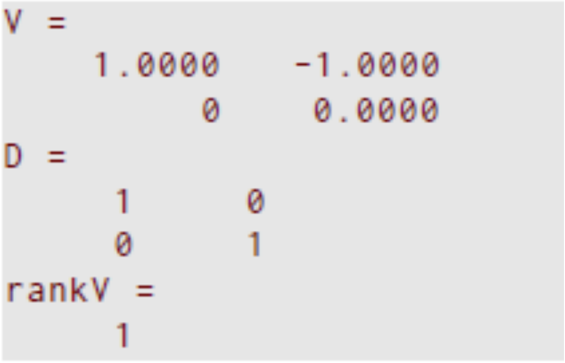
- What if A is not diagonalizable?

Computing EVD in Matlab

- If A is not diagonalizable, it will not have all eigenvalues distinct, and it will not have a full set of eigenvectors
- `eig` still works, but $\boldsymbol{V}^{-1}$ doesn't exist:
- $\boldsymbol{V}$ is only rank 1, indicating a single LI column
- You should get in the habit of critically evaluating output to check whether it is consistent with theory



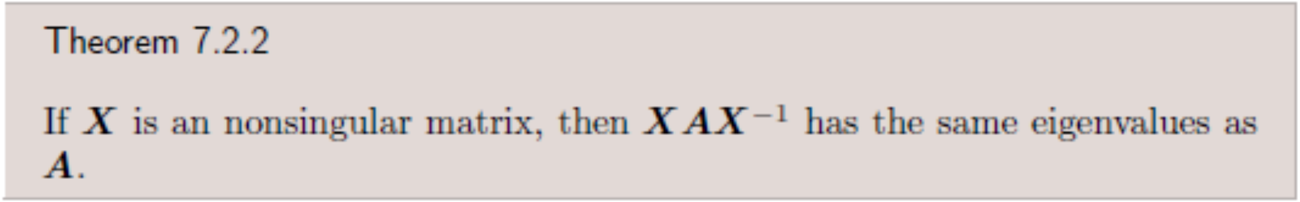
```
[V,D] = eig([1 1;0 1])
rankV = rank(V)
```



```
V =
    1.0000   -1.0000
         0    0.0000
D =
     1     0
     0     1
rankV =
     1
```

The EVD and similarity

- There is an important relationship between $\boldsymbol{A}$ and $\boldsymbol{D}$
- For any nonsingular matrix $\boldsymbol{S}$, then $\boldsymbol{B} = \boldsymbol{S}\boldsymbol{A}\boldsymbol{S}^{-1}$ is said to be similar to $\boldsymbol{A}$
- In many basic linear algebra texts, one can find a proof of this result:



Theorem 7.2.2

If $\boldsymbol{X}$ is an nonsingular matrix, then $\boldsymbol{X}\boldsymbol{A}\boldsymbol{X}^{-1}$ has the same eigenvalues as $\boldsymbol{A}$.

- There is a really nice interpretation of the similarity transformation

Similarity transformations

- Consider the produce of a nonsingular matrix $\boldsymbol{X}$ with any vector:
$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{z} = z_1\boldsymbol{x}_1 + \cdots + z_n\boldsymbol{x}_n$$
- The $\boldsymbol{x}_i$ are the columns of $\boldsymbol{X}$
- The columns of $\boldsymbol{X}$ are LI because it is invertible, so it is a basis for $\mathbb{C}^n$
- The z_i are the *coordinates* of $\boldsymbol{y}$ using the columns of $\boldsymbol{X}$ as a basis
- But, we could left multiply by $\boldsymbol{X}^{-1}$ and then $\boldsymbol{z} = \boldsymbol{X}^{-1}\boldsymbol{y}$
- The elements of $\boldsymbol{y}$ are now coordinates for $\boldsymbol{z}$ using the columns of $\boldsymbol{X}^{-1}$ as a basis
- Thus: *multiplication by the inverse of a matrix performs a* ***change of basis*** *into the coordinates associated with the matrix*

Similarity transformations

- Now consider the EVD
- Let $\boldsymbol{u} = \boldsymbol{A}\boldsymbol{x}$, or

$$\boldsymbol{u} = \boldsymbol{A}\boldsymbol{x} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^{-1}\boldsymbol{x}$$

- Premultiply by to get

$$\boldsymbol{V}^{-1}\boldsymbol{u} = \boldsymbol{D}(\boldsymbol{V}^{-1}\boldsymbol{x})$$

- This equation says that using the columns of $\boldsymbol{V}$ for a basis, that there is a diagonal relation between the two vectors $\boldsymbol{u}$ and $\boldsymbol{x}$
- Said another way, the EVD finds a basis for $\mathbb{C}^n$ so that the map is diagonal:

$$\boldsymbol{x} \mapsto \boldsymbol{A}\boldsymbol{x}$$

- In that case, each coordinate is just multiplied by its own scalar

Similarity transformations

- This can be really convenient for matrix powers; these will be an important operation for us
- Consider $\boldsymbol{A}^2$, and use the EVD:

$$\boldsymbol{A}^2 = (\boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^{-1})(\boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^{-1}) = \boldsymbol{V}\boldsymbol{D}(\boldsymbol{V}^{-1}\boldsymbol{V})\boldsymbol{D}\boldsymbol{V}^{-1} = \boldsymbol{V}\boldsymbol{D}^2\boldsymbol{V}^{-1},$$

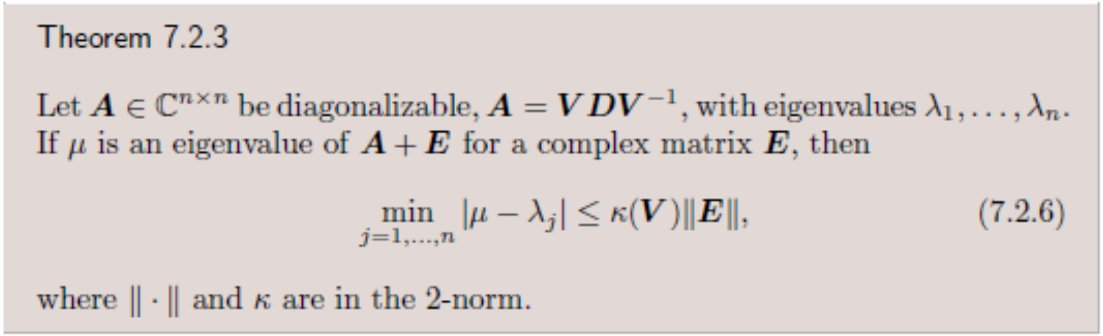
- If we new the EVD, we could just square the diagonal elements of $\boldsymbol{D}$, then reconstruct $\boldsymbol{A}^2$
- Higher powers? Then

$$\boldsymbol{A}^k = \boldsymbol{V}\boldsymbol{D}^k\boldsymbol{V}^{-1}$$

- Raise each eigenvalue to the *k*-th power to get $\boldsymbol{D}^k$!

Conditioning of EVD computation

- There are theorems around to tell us how much eigenvalues change in response to changes in the matrix
- One is the Bauer-Fike theorem:



Theorem 7.2.3

Let $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ be diagonalizable, $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^{-1}$, with eigenvalues $\lambda_1, \ldots, \lambda_n$. If μ is an eigenvalue of $\boldsymbol{A} + \boldsymbol{E}$ for a complex matrix $\boldsymbol{E}$, then

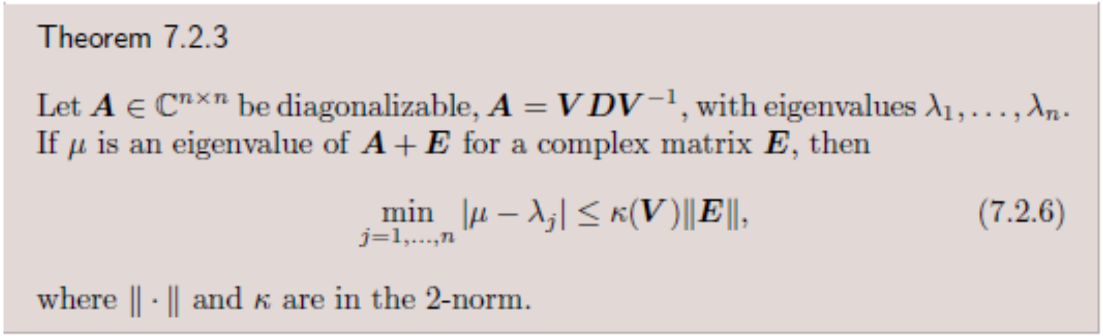
$$\min_{j=1,\ldots,n} |\mu - \lambda_j| \leq \kappa(\boldsymbol{V})\|\boldsymbol{E}\|, \tag{7.2.6}$$

where $\|\cdot\|$ and κ are in the 2-norm.

- This theorem says that small changes in the matrix can change the eigenvalues by as much as a factor of the condition number $\kappa(\boldsymbol{A})$
- If $\boldsymbol{A}$ is poorly conditioned, then this could be a significant change

Conditioning of EVD computation

- There are theorems around to tell us how much eigenvalues change in response to changes in the matrix
- One is the Bauer-Fike theorem:



Theorem 7.2.3

Let $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ be diagonalizable, $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^{-1}$, with eigenvalues $\lambda_1, \ldots, \lambda_n$. If μ is an eigenvalue of $\boldsymbol{A}+\boldsymbol{E}$ for a complex matrix $\boldsymbol{E}$, then

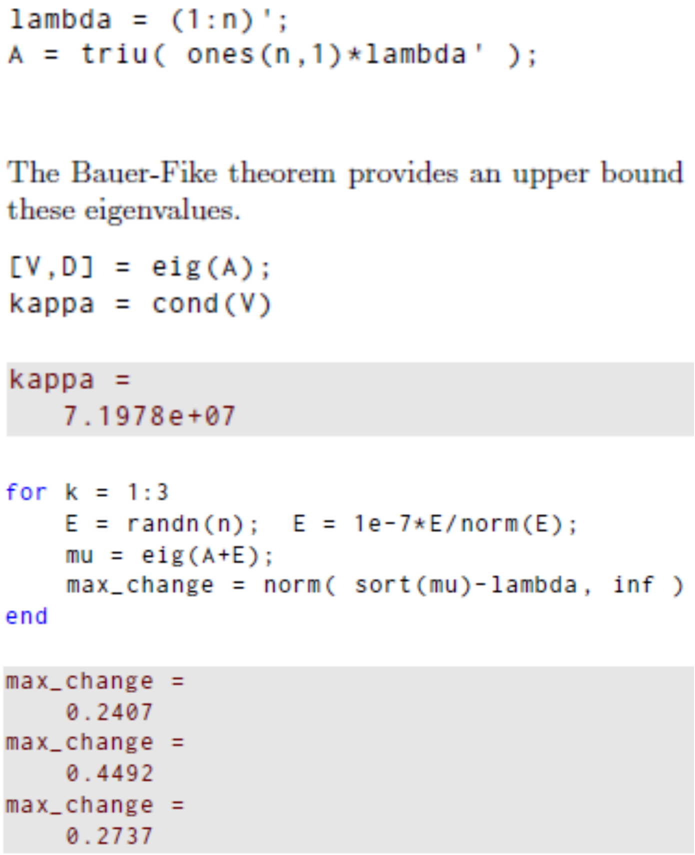
$$\min_{j=1,\ldots,n} |\mu - \lambda_j| \leq \kappa(\boldsymbol{V})\|\boldsymbol{E}\|, \qquad (7.2.6)$$

where $\|\cdot\|$ and κ are in the 2-norm.

- Eigenvalues by as much as a factor of the condition number $\kappa(\boldsymbol{V})$
- If $\boldsymbol{V}$ is unitary, $\kappa(\boldsymbol{V}) = 1$, $\boldsymbol{A}$ is *normal,* eigenvalue calculation robust
- If $\boldsymbol{V}$ nearly singular, $\kappa(\boldsymbol{V}) \gg 1$, eigenvalues can change significantly

Conditioning of EVD computation

- Example: triangular matrix with $n = 15$
- The bound on the change in the condition number is around 1e7
- Test the theorem with 1e-7 size perturbations
- The max change in these few cases is about 25% to 50% of the possible change



```
lambda = (1:n)';
A = triu( ones(n,1)*lambda' );
```

The Bauer-Fike theorem provides an upper bound these eigenvalues.

```
[V,D] = eig(A);
kappa = cond(V)
```

```
kappa =
   7.1978e+07
```

```
for k = 1:3
    E = randn(n);  E = 1e-7*E/norm(E);
    mu = eig(A+E);
    max_change = norm( sort(mu)-lambda, inf )
end
```

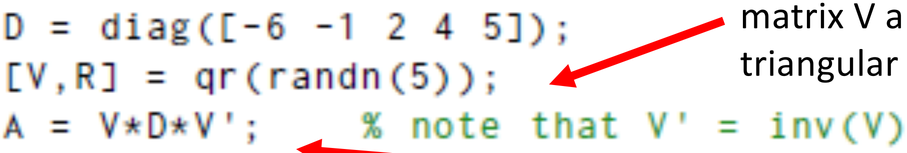
```
max_change =
    0.2407
max_change =
    0.4492
max_change =
    0.2737
```

Method of EVD computation

- The practical methods for computing the EVD are beyond the scope of this class (and book)
- It is worth pointing out that the methods often use the idea of matrix powers as part of the computation.
- If the eigenvalues are distinct, then raising them to a power separates them for large powers k
- There is an easy and elegant way to accomplish this separation

Method of EVD computation

- Example: create a matrix with known eigenvalues:

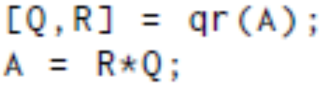


```
D = diag([-6 -1 2 4 5]);
[V,R] = qr(randn(5));
A = V*D*V';     % note that V' = inv(V)
```

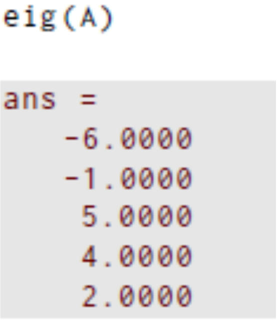
The qr function takes a random 5 by 5 matrix, then returns orthogonal matrix V and upper triangular matrix R

A and D are similar

- Now do QR factorization and reverse it:



```
[Q,R] = qr(A);
A = R*Q;
```

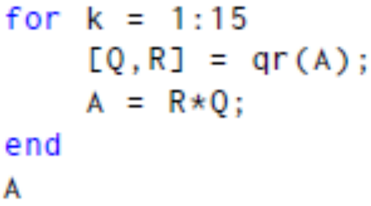


```
eig(A)

ans =
   -6.0000
   -1.0000
    5.0000
    4.0000
    2.0000
```

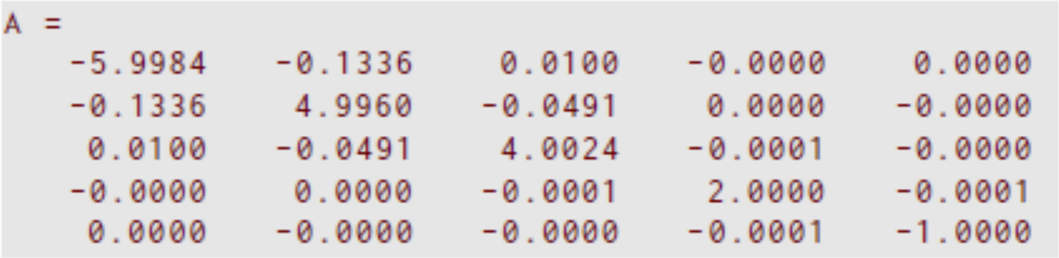
- We have same eigenvalues!

Method of EVD computation

- It turns out that we can repeat this and not change the eigenvalues!
- This is *Francis QR iteration*
- For this example:
- Look at diagonal elements, and off diagonal elements are getting small



```
for k = 1:15
    [Q,R] = qr(A);
    A = R*Q;
end
A
```

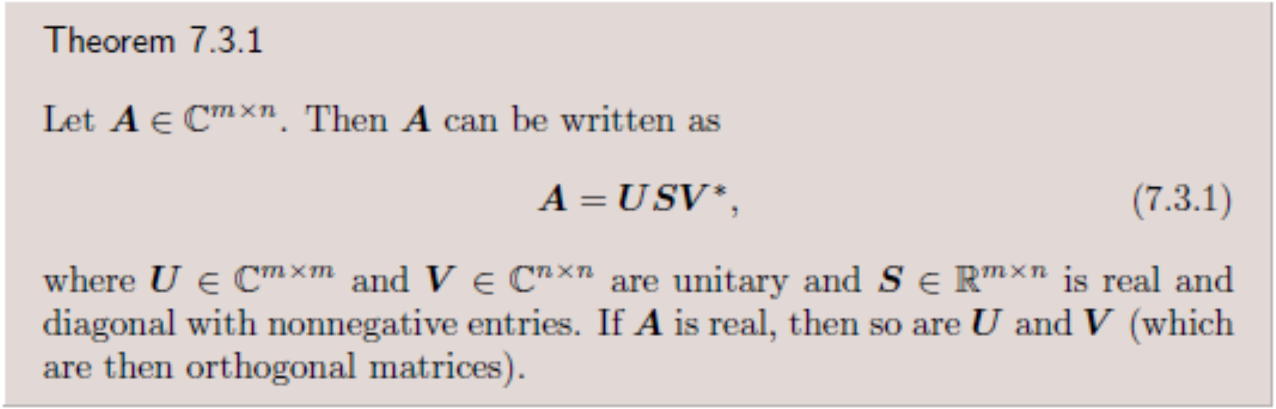


```
A =
   -5.9984   -0.1336    0.0100   -0.0000    0.0000
   -0.1336    4.9960   -0.0491    0.0000   -0.0000
    0.0100   -0.0491    4.0024   -0.0001   -0.0000
   -0.0000    0.0000   -0.0001    2.0000   -0.0001
    0.0000   -0.0000   -0.0000   -0.0001   -1.0000
```

Section 7.3
Singular value decomposition

Singular value decomposition

- Here is another matrix factorization; it is widely used in many fields:



Theorem 7.3.1

Let $\boldsymbol{A} \in \mathbb{C}^{m\times n}$. Then $\boldsymbol{A}$ can be written as

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^*, \tag{7.3.1}$$

where $\boldsymbol{U} \in \mathbb{C}^{m\times m}$ and $\boldsymbol{V} \in \mathbb{C}^{n\times n}$ are unitary and $\boldsymbol{S} \in \mathbb{R}^{m\times n}$ is real and diagonal with nonnegative entries. If $\boldsymbol{A}$ is real, then so are $\boldsymbol{U}$ and $\boldsymbol{V}$ (which are then orthogonal matrices).

- This is the singular value decomposition, or SVD
- Cols of U: left singular vectors; cols of V: right singular vectors
- Diagonal elements of S are the singular values $\sigma_1, \dots, \sigma_r, r = \min\{m, n\}$

Singular value decomposition

- The SVD is
$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^*$$
- $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ is $m \times n$, with complex entries
- Columns of (unitary) $\boldsymbol{U} \in \mathbb{C}^{m \times m}$: left singular vectors
- Columns of (unitary) $\boldsymbol{V} \in \mathbb{C}^{n \times n}$: right singular vectors
- Diagonal elements of $\boldsymbol{S} \in \mathbb{R}^{m \times n}$ are the singular values
$$\sigma_1, \ldots, \sigma_r, r = \min\{m, n\}$$
- Usually the ordering of the singular values is
$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$$
- σ_1 is the principal singular value
- $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$ are the principal singular vectors

Singular value decomposition

- What about real-valued $\boldsymbol{A}$?
- Consider the case with $\boldsymbol{A} \in \mathbb{R}^{m \times n}$

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\boldsymbol{T}}$$

- Columns of orthogonal $\boldsymbol{U} \in \mathbb{R}^{m \times m}$: left singular vectors
- Columns of unitary $\boldsymbol{V} \in \mathbb{R}^{n \times n}$: right singular vectors
- Diagonal elements of $\boldsymbol{S} \in \mathbb{R}^{m \times n}$ are the singular values

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0, \qquad r = \min\{m, n\}$$

- This is still an SVD because it satisfies the requirements:
  - the first and last matrices are orthogonal,
  - and the middle matrix is diagonal with non negative entries

Interpreting the SVD

- Let's rewrite the SVD

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^*$$

- As follows (postmultiply by V):

$$\boldsymbol{A}\boldsymbol{V} = \boldsymbol{U}\boldsymbol{S}$$

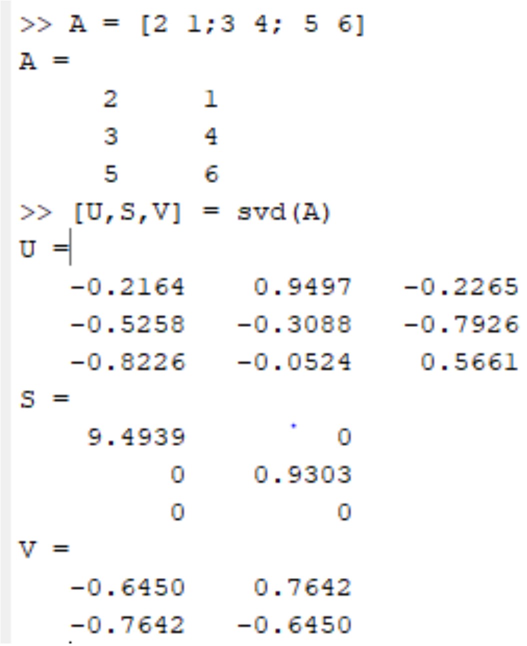
- Considering one column at a time, we have

$$\boldsymbol{A}\boldsymbol{v}_k = \sigma_k \boldsymbol{u}_k, \qquad k = 1,2, \dots, r, \qquad r = \min\{m, n\}$$

- This means that each right singular vector $\boldsymbol{v}_k$ is mapped by $\boldsymbol{A}$ to a scalar multiple (σ_k) of the corresponding left singular vector $\boldsymbol{u}_k$

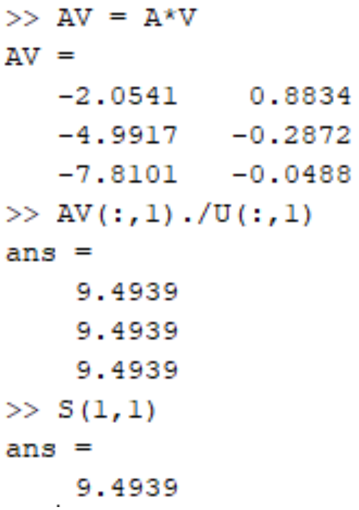
Interpreting the SVD: example

- Compute an SVD:



```
>> A = [2 1;3 4; 5 6]
A =
     2     1
     3     4
     5     6
>> [U,S,V] = svd(A)
U =
   -0.2164    0.9497   -0.2265
   -0.5258   -0.3088   -0.7926
   -0.8226   -0.0524    0.5661
S =
    9.4939         0
         0    0.9303
         0         0
V =
   -0.6450    0.7642
   -0.7642   -0.6450
```

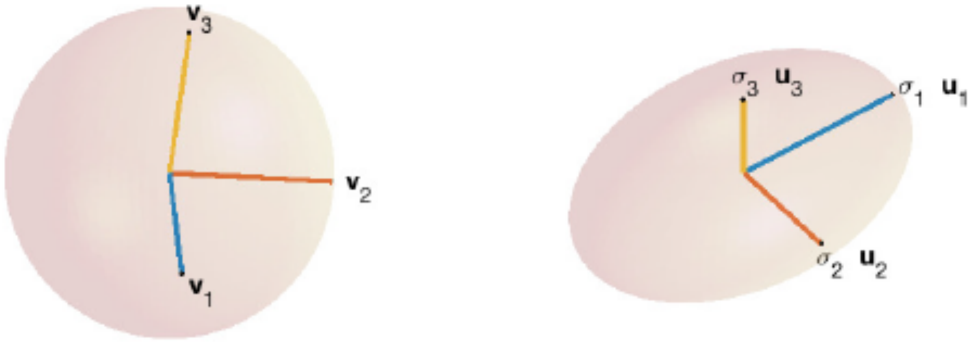
- Compare the first column of $\boldsymbol{AV}$ with the first column of $\boldsymbol{U}$: this should be just σ_1



```
>> AV = A*V
AV =
   -2.0541    0.8834
   -4.9917   -0.2872
   -7.8101   -0.0488
>> AV(:,1)./U(:,1)
ans =
    9.4939
    9.4939
    9.4939
>> S(1,1)
ans =
    9.4939
```

Interpreting the SVD: visualizing it

- The SVD can be visualized for 3x3 real-valued matrices
- Unit vectors in the directions of the columns of V (left) are distorted by multiplication by A (right) to



Contrasting the EVD and the SVD

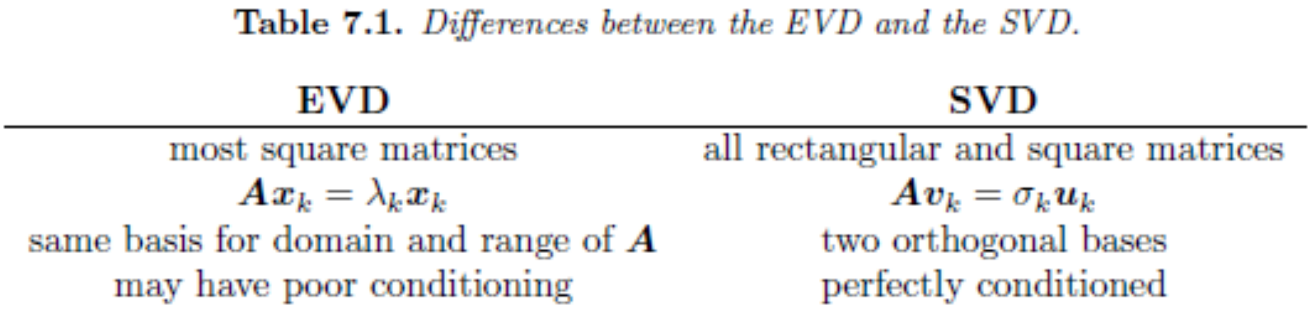


Table 7.1. *Differences between the EVD and the SVD.*

EVD	**SVD**
most square matrices	all rectangular and square matrices
$\boldsymbol{A}\boldsymbol{x}_k = \lambda_k \boldsymbol{x}_k$	$\boldsymbol{A}\boldsymbol{v}_k = \sigma_k \boldsymbol{u}_k$
same basis for domain and range of $\boldsymbol{A}$	two orthogonal bases
may have poor conditioning	perfectly conditioned

Connection between the SVD and 2-norm (Thrm 7.3.2)

- Let $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ have an SVD with $\boldsymbol{A} = \boldsymbol{USV}^*$ and
$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0, r = \min\{m, n\}$$
- Then:

1. The 2-norm satisfies $\left|\left|\boldsymbol{A}\right|\right|_2 = \sigma_1$.
2. The rank of A is the number of nonzero singular values.
3. The condition number satisfies
$$\kappa(\boldsymbol{A}) = \left|\left|\boldsymbol{A}\right|\right|_2\left|\left|\boldsymbol{A}^+\right|\right|_2 = \sigma_1/\sigma_r.$$

Division by zero here implies that $\boldsymbol{A}$ does not have full rank.
Recall that $\boldsymbol{A}^+ = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T$ is the pseudoinverse (chapter 3).

SVD example 7.3.2: try it

```
A=vander(1:5);          % Vandermonde matrix
A = A(:,1:4)            % 5 by 4 now
[U,S,V] = svd(A);
norm(U’*U-eye(5))       % check U is orthogonal
norm(V’*V-eye(4))       % check V is orthogonal
sigma = diag(S)
[norm(A) sigma(1)]                      % Thrm 7.3.2, no. 1
[cond(A) sigma(1)/sigma(end)]           % Thrm 7.3.2, no. 3
```

Connections between SVD and EVD

- Let $\boldsymbol{A} = \boldsymbol{USV}^*$
- Create the square hermitian matrix $\boldsymbol{B} = \boldsymbol{A}^*\boldsymbol{A}$
- Then

$$\boldsymbol{B} = (\boldsymbol{V}\boldsymbol{S}^*\boldsymbol{U}^*)(\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^*) = \boldsymbol{V}\boldsymbol{S}^*\boldsymbol{S}\boldsymbol{V}^* = \boldsymbol{V}(\boldsymbol{S}^T\boldsymbol{S})\boldsymbol{V}^{-1}.$$

- $\boldsymbol{S}^T\boldsymbol{S}$ is a diagonal $n \times n$ matrix
- There are two cases:

$\underline{m \geq n:}$

$$\boldsymbol{S}^T\boldsymbol{S} = \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_n^2 \end{bmatrix},$$

$\underline{m < n:}$

$$\boldsymbol{S}^T\boldsymbol{S} = \begin{bmatrix} \sigma_1^2 & & & \\ & \ddots & & \\ & & \sigma_m^2 & \\ & & & \boldsymbol{0} \end{bmatrix},$$

- The lower right zero is $n - m \times n - m$

Connections between SVD and EVD

- There are two cases:

$\underline{m \geq n:}$

$$\boldsymbol{S}^T\boldsymbol{S} = \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_n^2 \end{bmatrix},$$

$\underline{m < n:}$

$$\boldsymbol{S}^T\boldsymbol{S} = \begin{bmatrix} \sigma_1^2 & & & \\ & \ddots & & \\ & & \sigma_m^2 & \\ & & & \mathbf{0} \end{bmatrix},$$

- The squares of the singular values of $\boldsymbol{A}$ are the eigenvalues of $\boldsymbol{B} = \boldsymbol{A}^*\boldsymbol{A}$!
- Conversely, the EVD of $\boldsymbol{B}$ yields the singular values of $\boldsymbol{A}$ and the right singular vectors of $\boldsymbol{A}$
- We could get the left singular vectors from $\boldsymbol{AV} = \boldsymbol{US}$ by doing one column at a time

Connections between SVD and EVD

- We could also create

which is $m+n \times m+n$

for $\boldsymbol{A}$ $m \times n$

$$C = \begin{bmatrix} 0 & A^* \\ A & 0 \end{bmatrix}.$$

- If σ is a singular value of $\boldsymbol{B}$, the σ and $-\sigma$ are eigenvalues of $\boldsymbol{C}$
- The associated eigenvector immediately gives a left and right singular vector of $\boldsymbol{A}$
- This connection is implicitly exploited by software to compute the SVD

Thin form of the SVD

- Like the QR factorization, we can have both full and thin versions of the factorization
- Let $\boldsymbol{A} = \boldsymbol{USV}^*$ where $\boldsymbol{A}$ is $m \times n$ with $m > n$
- Then we can write

$$US = \begin{bmatrix} u_1 & \cdots & u_n & u_{n+1} & \cdots & u_m \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ & \mathbf{0} & \\ & & \end{bmatrix}$$

Last $m - n$ rows are zero here

$$= \begin{bmatrix} u_1 & \cdots & u_n \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} = \hat{U} \hat{S},$$

$\widehat{\boldsymbol{U}}$ is $m \times n$, and
$\widehat{\boldsymbol{S}}$ is $n \times n$, but still
same info about $\boldsymbol{A}$!

- Use svd(A,0) to get this in Matlab

Section 7.4
Symmetry and definiteness

Symmetry and Definiteness

- When we had real matrices $\boldsymbol{A}$, we found in chapter 2 that there could be specializations of the factorizations there: e.g., $\boldsymbol{A} = \boldsymbol{LDL}^T$, where $\boldsymbol{D}$ is diagonal and $\boldsymbol{L}$ is lower triangular
- We now study the analogous hermitian case for $\boldsymbol{A}^* = \boldsymbol{A}$
- Let $\boldsymbol{A} = \boldsymbol{USV}^*$ where $\boldsymbol{A}$ is $n \times n$
- Since $\boldsymbol{S}$ is real and square,

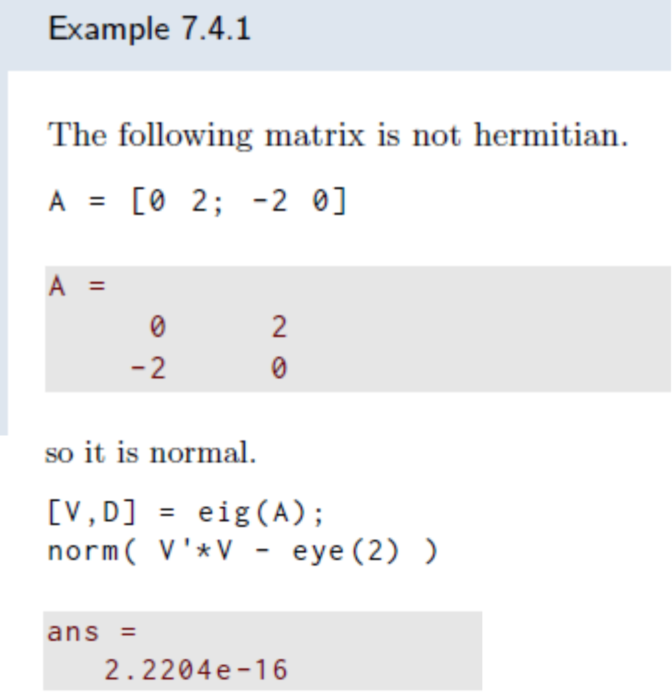
$$\boldsymbol{A}^* = \boldsymbol{VS}^*\boldsymbol{U}^* = \boldsymbol{VSU}^*$$

- In this case $\boldsymbol{A}$ is diagonalizable with

$$\boldsymbol{A} = \boldsymbol{VDV}^{-1} = \boldsymbol{VDV}^*$$

- Here $\boldsymbol{V}$ is unitary, and $\boldsymbol{D}$ is diagonal and real
- This means that a hermitian matrix has a complete set of orthonormal eigenvectors (a unitary diagonalization), with real eigenvalues

Symmetry and Definiteness: example



Example 7.4.1

The following matrix is not hermitian.

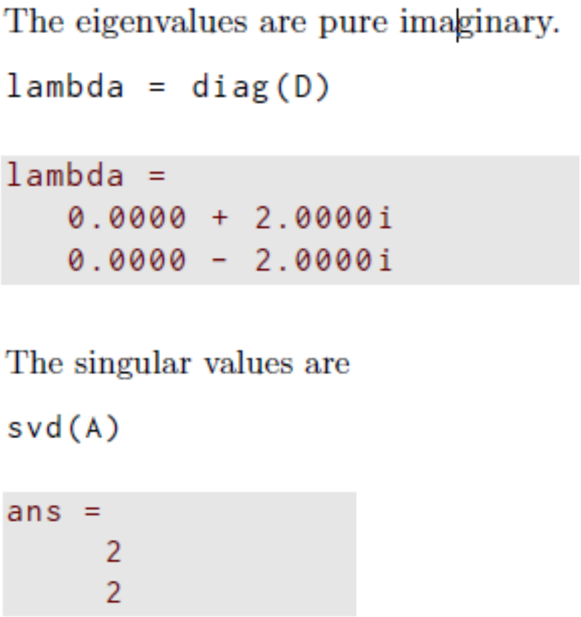
```
A = [0 2; -2 0]
```

```
A =
     0     2
    -2     0
```

so it is normal.

```
[V,D] = eig(A);
norm( V'*V - eye(2) )
```

```
ans =
   2.2204e-16
```



The eigenvalues are pure imaginary.

```
lambda = diag(D)
```

```
lambda =
   0.0000 + 2.0000i
   0.0000 - 2.0000i
```

The singular values are

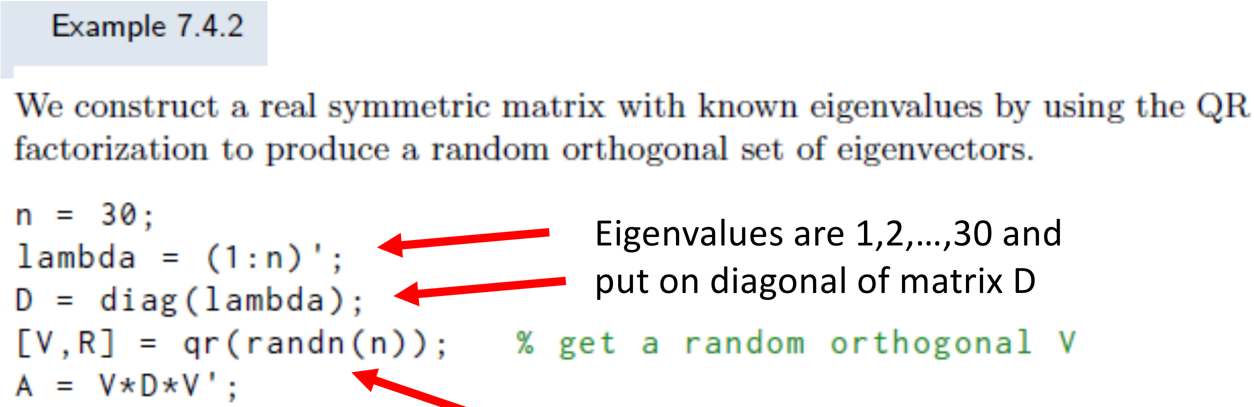
```
svd(A)
```

```
ans =
     2
     2
```

Symmetry and Definiteness

- Now we have a theorem that says that the condition number for the eigenvalues is bounded above by $\kappa(\boldsymbol{V})$
- Here $\boldsymbol{V}$ is the eigenvector matrix
- But, it is unitary (or orthogonal), which means that $\kappa = 1$!
- So, our last theorem implies that the condition number for a Hermitian or normal matrix is one, which is as good as it gets!
- In that case, that is for Hermitian or normal matrices, the eigenvalues can be changed by no more than the norm of the perturbation of the matrix!
- Can we verify this?

Symmetry and Definiteness: example



Example 7.4.2

We construct a real symmetric matrix with known eigenvalues by using the QR factorization to produce a random orthogonal set of eigenvectors.

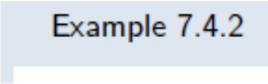
```
n = 30;
lambda = (1:n)';
D = diag(lambda);
[V,R] = qr(randn(n));    % get a random orthogonal V
A = V*D*V';
```

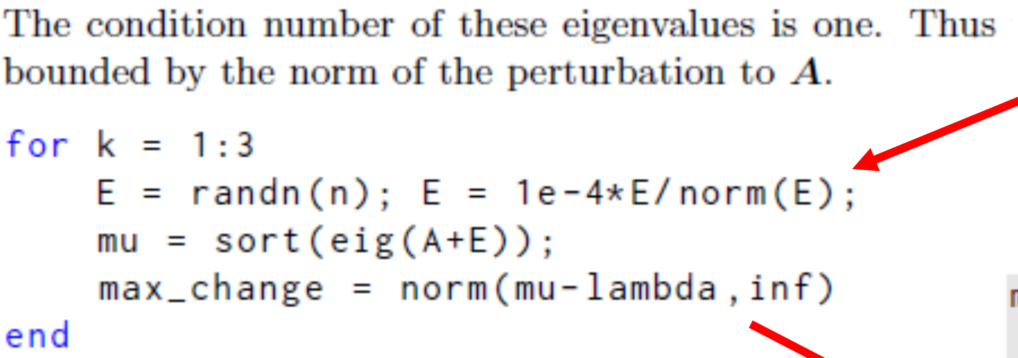
Eigenvalues are 1,2,…,30 and put on diagonal of matrix D

The qr function takes a random 30 by 30 matrix here, then returns an orthogonal matrix V and upper triangular matrix R

The orthonormal columns of V become the associated eigenvectors for the eigenvalues in matrix D, which are the same for A. Why?

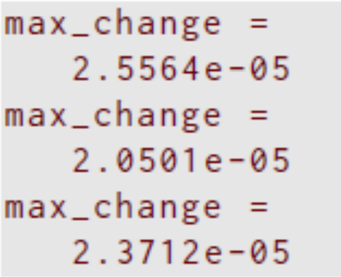
Symmetry and Definiteness: example


Example 7.4.2


The condition number of these eigenvalues is one. Thus bounded by the norm of the perturbation to $\boldsymbol{A}$.

```
for k = 1:3
    E = randn(n); E = 1e-4*E/norm(E);
    mu = sort(eig(A+E));
    max_change = norm(mu-lambda,inf)
end
```

Create perturbation matrix E that is random and has norm 1e-4


```
max_change =
   2.5564e-05
max_change =
   2.0501e-05
max_change =
   2.3712e-05
```

Symmetry and Definiteness

- This is great, but it is not quite an SVD
- Why? The sign of the diagonal elements could be anything.
- Can we make it into an SVD? Yes!
- The trouble is with $\boldsymbol{D}$; lets rewrite it as a product of two diagonal matrices: one is $\boldsymbol{T}$ with the $\text{sign}(d_{ii})$ on the diagonal; the other is $|\boldsymbol{D}|$, which has $|d_{ii}|$ on the diagonal. We still have
$$\boldsymbol{D} = \boldsymbol{T}|\boldsymbol{D}|$$
- Substitute to get
$$\boldsymbol{A} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^* = \boldsymbol{V}\boldsymbol{T}|\boldsymbol{D}|\boldsymbol{V}^* = (\boldsymbol{V}\boldsymbol{T})|\boldsymbol{D}|(\boldsymbol{V}^*)$$
- This *is* an SVD, because the diagonal matrix has nonnegative entries and the left and right matrices are unitary

Symmetry and definiteness

- Recall the quadratic form $\boldsymbol{x}^*\boldsymbol{A}\boldsymbol{x}$ where $\boldsymbol{A}$ is $n\times n$ and $\boldsymbol{x}$ is $n\times 1$.
- For real entries, $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$ is the quadratic form of interest
- For n=2, we can easily write it out:

$$[x_1 \quad x_2]\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}=a_{11}x_1^2+(a_{12}+a_{21})x_1x_2+a_{22}x_2^2$$

- For some matrices, this quantity remains positive for any nonzero vector $\boldsymbol{x}$
- For example, if $a_{11}>0, a_{22}>0$, and $a_{21}=-a_{12}$, then the quadratic form reduces to $a_{11}x_1^2+a_{22}x_2^2>0$ for nonzero $\boldsymbol{x}$
- This matrix is called positive definite
- If the matrix were symmetric and this were true, then it is symmetric positive definite or SPD

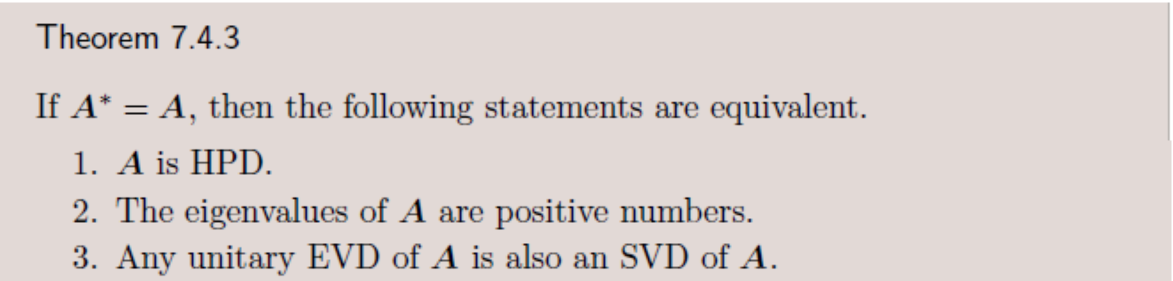
Symmetry and definiteness

- The SPD matrix is important: it has all positive eigenvalues
- Complex case: a matrix is Hermitian positive definite (HPD) when

$$\boldsymbol{A} = \boldsymbol{A}^* \text{ and } \boldsymbol{x}^*\boldsymbol{A}\boldsymbol{x} > \boldsymbol{0}$$

for any nonzero compatible $\boldsymbol{x}$

- It turns out that we can prove some equivalent statements:



Theorem 7.4.3

If $\boldsymbol{A}^* = \boldsymbol{A}$, then the following statements are equivalent.

1. $\boldsymbol{A}$ is HPD.
2. The eigenvalues of $\boldsymbol{A}$ are positive numbers.
3. Any unitary EVD of $\boldsymbol{A}$ is also an SVD of $\boldsymbol{A}$.

- This means, e.g., that all positive eigenvalues implies HPD (or SPD)

Rayleigh Quotient and eigenvalues

- In your linear algebra class, you may have discussed the Rayleigh quotient
- For $\boldsymbol{A}$ $n \times n$ and $\boldsymbol{x}$ $n \times 1$ with complex entries, the Rayleigh quotient is

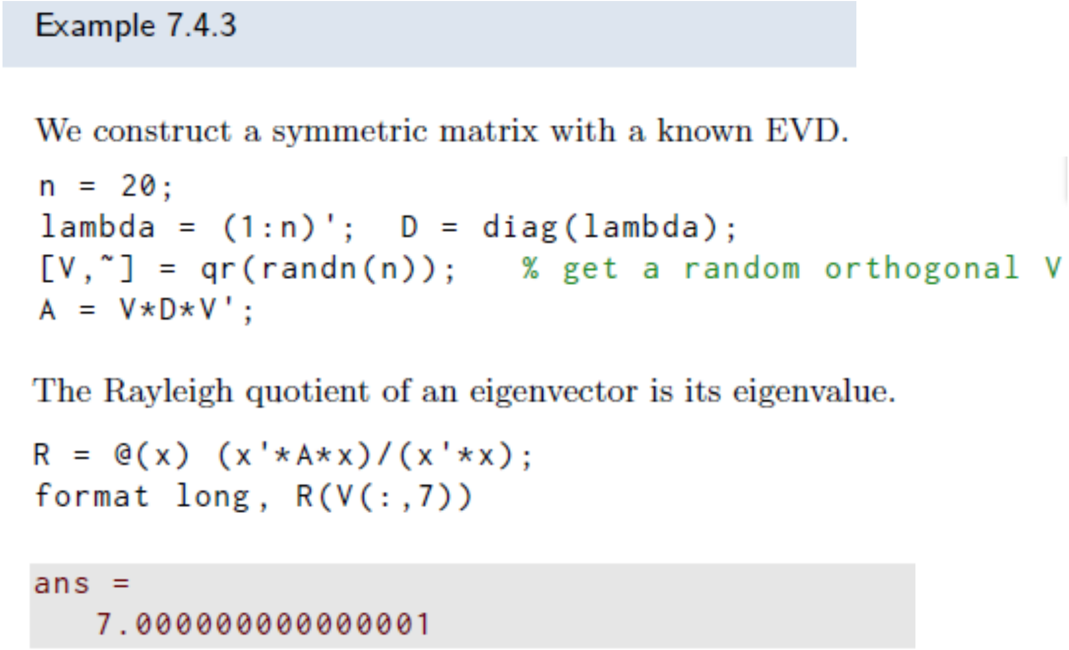
$$R_A(\boldsymbol{x}) = \frac{\boldsymbol{x}^* \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^* \boldsymbol{x}}$$

- In the special case that x is an eigenvector, say $\boldsymbol{v}$, then $\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}$, and substitution easily gives that $R_A(\boldsymbol{v}) = \lambda$
- We can conclude that: *the Rayleigh quotient maps an eigenvector into its associated eigenvalue*

Rayleigh Quotient and eigenvalues

- Consider the Hermitian case $\boldsymbol{A} = \boldsymbol{A}^*$ and the Rayleigh quotient
$$R_A(\boldsymbol{x}) = \frac{\boldsymbol{x}^* \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^* \boldsymbol{x}}$$
- Put in a vector that is close to an eigenvector: $\boldsymbol{x} = \boldsymbol{v} + \epsilon \boldsymbol{z}$ for $\epsilon \to 0$
- Using a multidimensional Taylor expansion gives that
$$R_A(\boldsymbol{v} + \epsilon \boldsymbol{z}) = R_A(v) + 0 + O(\epsilon^2) = \lambda + O(\epsilon^2)$$
- This happens because, for an eigenvector, $\nabla R_A(\boldsymbol{v}) = \boldsymbol{0}$
- This means that if the input vector is within $O(\epsilon)$ of an eigenvector then the Rayleigh quotient is within $O(\epsilon^2)$ of the eigenvalue: R_A does a good job of approximating the eigenvalue! Let's explore this…

Rayleigh Quotient and eigenvalues



Example 7.4.3

We construct a symmetric matrix with a known EVD.

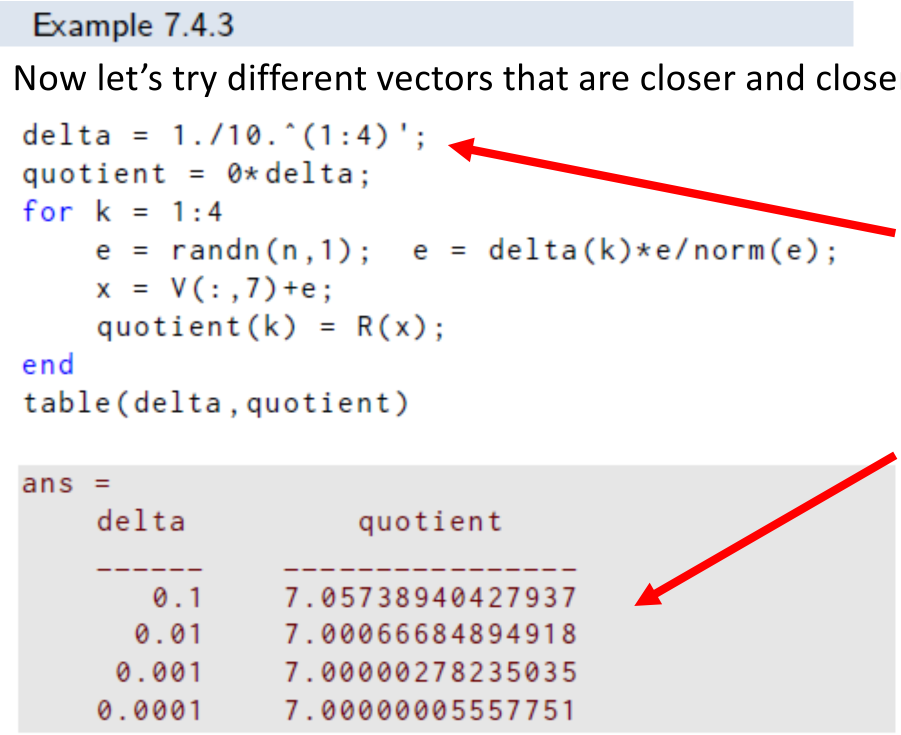
```
n = 20;
lambda = (1:n)';  D = diag(lambda);
[V,~] = qr(randn(n));   % get a random orthogonal V
A = V*D*V';
```

The Rayleigh quotient of an eigenvector is its eigenvalue.

```
R = @(x) (x'*A*x)/(x'*x);
format long, R(V(:,7))
```

```
ans =
   7.000000000000001
```

Rayleigh Quotient and eigenvalues



Example 7.4.3

Now let’s try different vectors that are closer and closer to an eigenvector

```
delta = 1./10.^(1:4)';
quotient = 0*delta;
for k = 1:4
    e = randn(n,1);  e = delta(k)*e/norm(e);
    x = V(:,7)+e;
    quotient(k) = R(x);
end
table(delta,quotient)
```

```
ans =
    delta          quotient    
    ______    ________________

       0.1    7.05738940427937
      0.01    7.00066684894918
     0.001    7.00000278235035
    0.0001    7.00000005557751
```

Every time the input vector gets a factor of 10 closer to the eigenvector, there is a factor of 100 improvement in the eigenvalue approximation (another two zeros after the decimal point)

Section 7.5
Dimension reduction

Dimension Reduction

- We now return to the SVD
- We want to use it to approximate the information in a matrix with much less storage, or many less numbers
- We will see a few examples of this
- Consider matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ with $m > n$ (for now)
- The thin SVD is then

$$A = \widehat{U}\widehat{S}V^T = \begin{bmatrix} u_1 & u_2 & \cdots & u_n \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_n^T \end{bmatrix}$$

- We truncated the n+1 to m columns of $\boldsymbol{U}$ and rows of $\boldsymbol{S}$ to get $\widehat{\boldsymbol{U}}, \widehat{\boldsymbol{S}}$

Dimension Reduction

- Now let’s rewrite the thin SVD *carefully*
- Note that $\boldsymbol{\widehat{U}}$ has orthonormal *columns* $\boldsymbol{u}_i$
- $\boldsymbol{V}$ has orthonormal *rows* $\boldsymbol{v}_i^T$

$$A = \widehat{U}\widehat{S}V^T = \begin{bmatrix} u_1 & u_2 & \cdots & u_n \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_n^T \end{bmatrix}$$

$n \times n$ here

$m \times n$ here

$$= \begin{bmatrix} \sigma_1 u_1 & \cdots & \sigma_n u_n \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_n^T \end{bmatrix}$$

$$= \sigma_1 u_1 v_1^T + \cdots + \sigma_r u_r v_r^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T,$$

$u_i v_i^T$ is an *outer product* here, with each forming an $m \times n$ matrix

Dimension Reduction

- Each of those outer products is weighed with the singular value, then adding them *all* up recovers the original matrix

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T,$$

- But, it so happens that in many matrices, there are only a few singular values that are sizable, and the rest may be quite small.
- By convention we ordered the singular values:

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$$

- So, if the largest few singular values are say $1,2, \ldots, k$, then maybe we need to keep only the first k terms to get a good approximation to the content of $\boldsymbol{A}$!
- Let's make this more precise.

Dimension Reduction

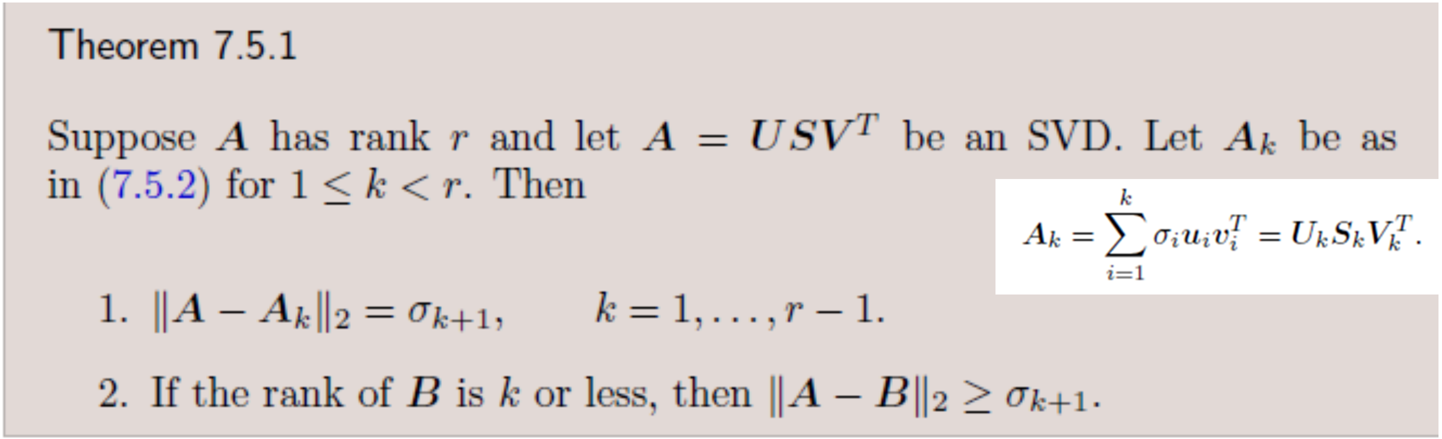
- We can think of the process of keeping more singular values as generating a sequence of matrices for increasing k, with $1 \leq k \leq r$
- Then, we only do a partial sum using the first k terms:

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T = U_k S_k V_k^T .$$

- $\boldsymbol{U}_k$ and $\boldsymbol{V}_k$ are the first k columns of $\boldsymbol{U}$ and $\boldsymbol{V}$
- $\boldsymbol{S}_k$ is the upper left $k \times k$ submatrix of $\boldsymbol{S}$
- Because each $u_i v_i^T$ is a matrix with unit norm, the overall size of each matrix added is given by the singular value
- Because of this, there are many cases where stopping at a small value of k relative to r will give a v

Dimension Reduction

- What do we know about the $\boldsymbol{A}_k$?
- The rank of a sum of matrices is less than or equal to the sum of the ranks of each: thus, the rank of $\boldsymbol{A}_k$ is at most k
- And, it turns out that $\boldsymbol{A}_k$ is the *best rank k approximation* to A!



Theorem 7.5.1

Suppose A has rank r and let $A = USV^T$ be an SVD. Let A_k be as in (7.5.2) for $1 \leq k < r$. Then

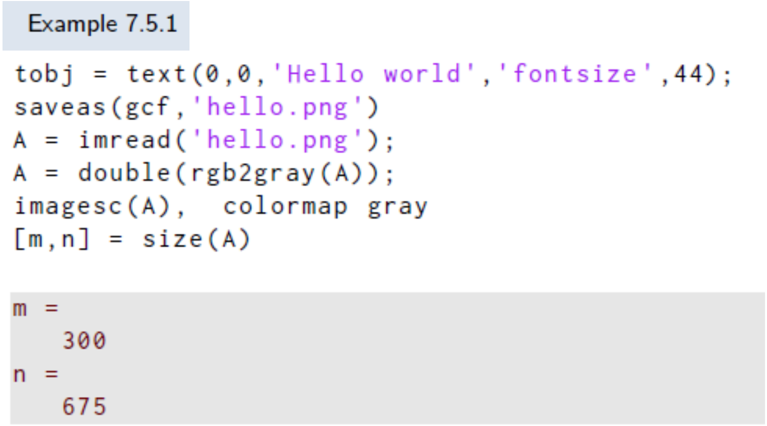
$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T = U_k S_k V_k^T .$$

1. $\|A - A_k\|_2 = \sigma_{k+1}, \qquad k = 1, \ldots, r-1.$
2. If the rank of B is k or less, then $\|A - B\|_2 \geq \sigma_{k+1}$.

Dimension Reduction: example

- For many matrices, k need not be very large to get a good approximation to the original matrix
- Example 7.5.1 gives an example with text
- Demo of that example and/or others…

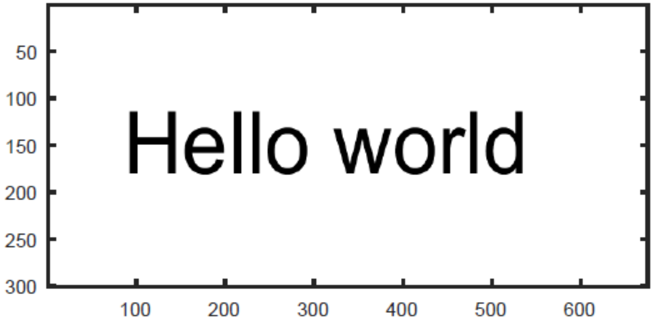
Dimension Reduction: example

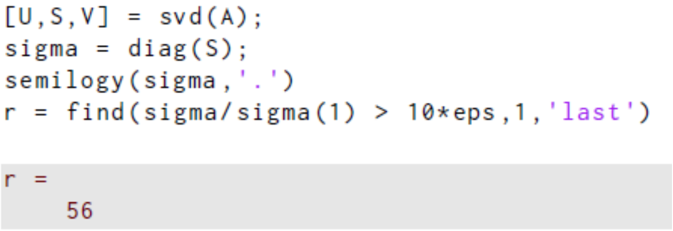


Example 7.5.1

```
tobj = text(0,0,'Hello world','fontsize',44);
saveas(gcf,'hello.png')
A = imread('hello.png');
A = double(rgb2gray(A));
imagesc(A),  colormap gray
[m,n] = size(A)
```

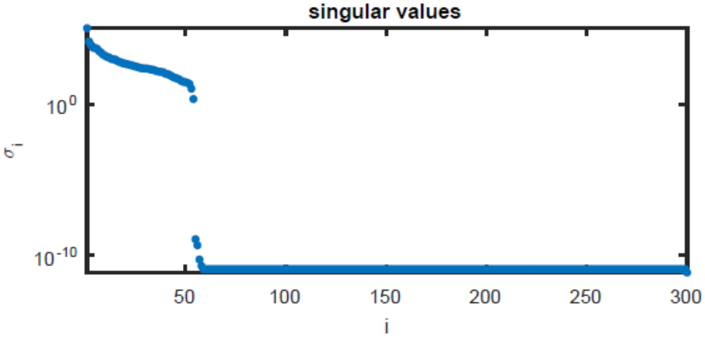
```
m =
   300
n =
   675
```



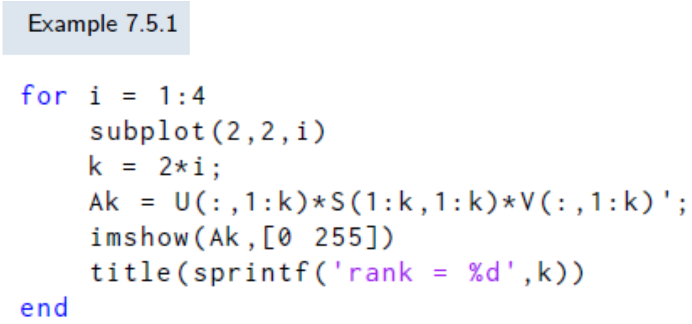


```
[U,S,V] = svd(A);
sigma = diag(S);
semilogy(sigma,'.')
r = find(sigma/sigma(1) > 10*eps,1,'last')
```

```
r =
   56
```

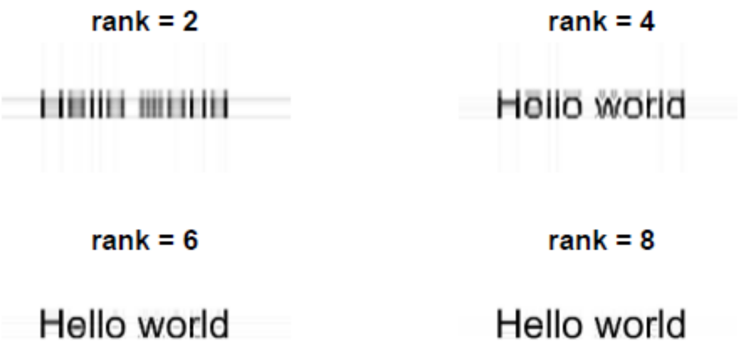


Dimension Reduction: example



Example 7.5.1

```
for i = 1:4
    subplot(2,2,i)
    k = 2*i;
    Ak = U(:,1:k)*S(1:k,1:k)*V(:,1:k)';
    imshow(Ak,[0 255])
    title(sprintf('rank = %d',k))
end
```



Look how few singular values are needed to get a decent looking image! Less than 5% of the original storage to get the rank 8 approximation!

Dimension Reduction: example

- These low rank approximations can be used to get at the essence of data
- One measure of how much content is contained in each added rank is the fractional "energy" given by

$$\tau_k = \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{r} \sigma_i^2}, \quad k = 1, \ldots, r.$$

- Consider a different matrix of information now: Example 7.5.2
- Here we look at the voting pattern for the Senate in the 111th session of the US Congress

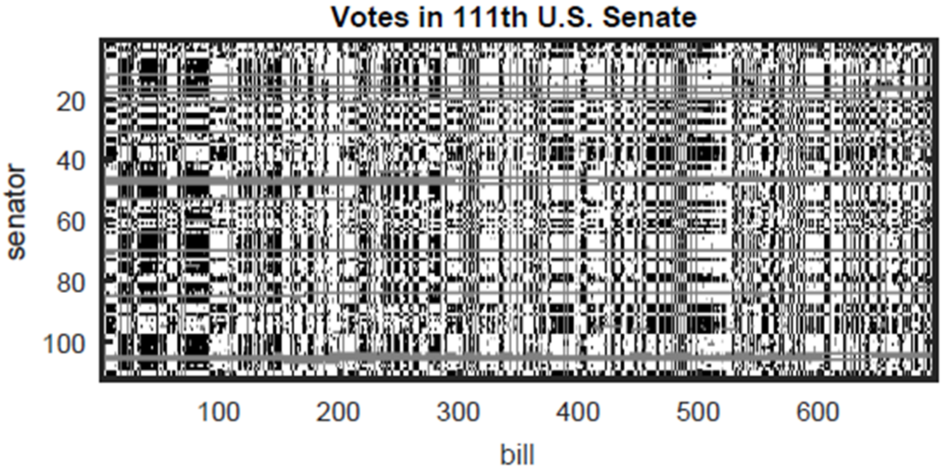
Dimension Reduction: example

Example 7.5.2

```
load voting    % get from the book's website
[m,n] = size(A);
```

If we visualize the votes (white is “yea,” black is “nay,” and gray is anything else), we can see great similarity between many rows, reflecting party unity.

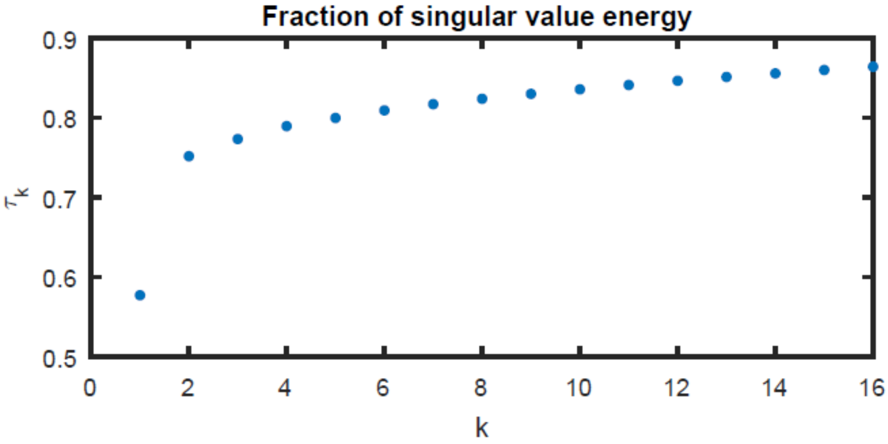
```
imagesc(A)
colormap gray
```



Dimension Reduction: example

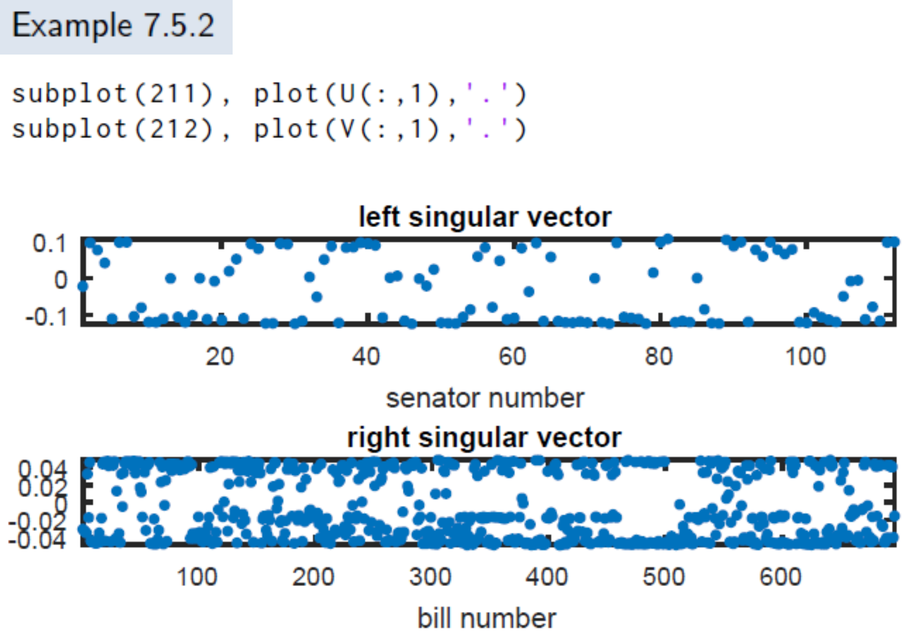
Example 7.5.2

```
[U,S,V] = svd(A);
sigma = diag(S);
tau = cumsum(sigma.^2) / sum(sigma.^2);
plot(tau(1:16),'.')
```



- The first two singular values account for 75% of the energy
- The remaining ones each account for relatively little of the energy
- Maybe just those first two are enough to capture the essence of the data?

Dimension Reduction: example

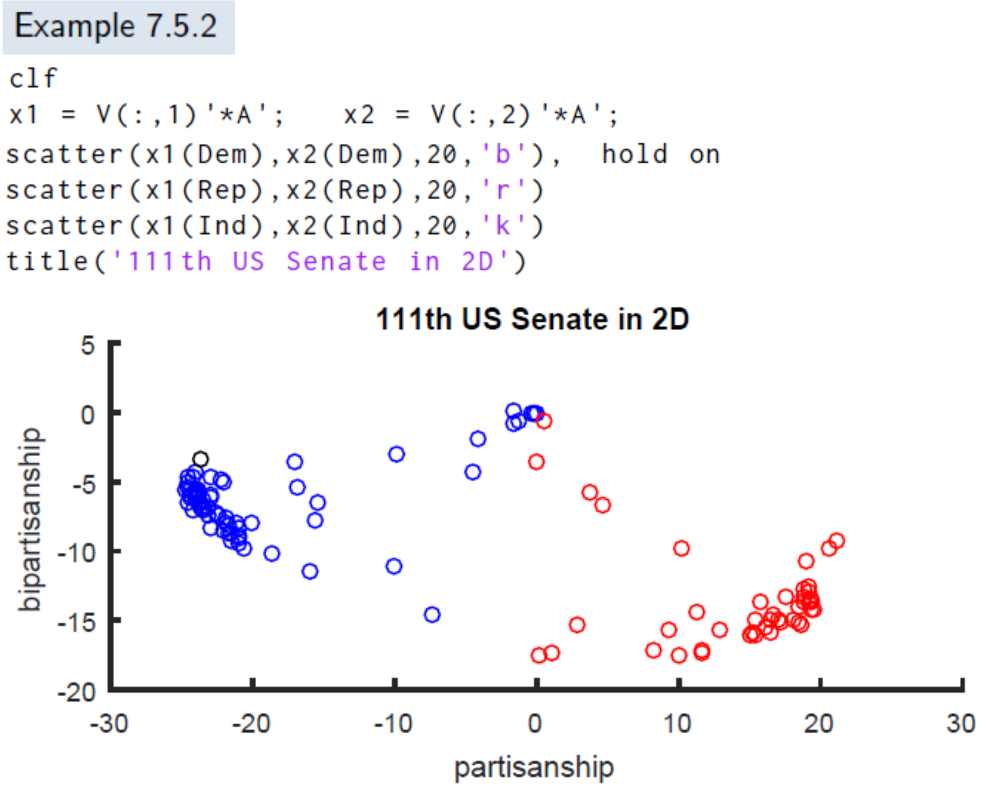


Example 7.5.2

```
subplot(211), plot(U(:,1),'.')
subplot(212), plot(V(:,1),'.')
```

- Note the different sizes of the two vectors
- Most of the values in these vectors are at $\pm C$: there is not much in the middle

Dimension Reduction: example



Example 7.5.2

```
clf
x1 = V(:,1)'*A';   x2 = V(:,2)'*A';
scatter(x1(Dem),x2(Dem),20,'b'),  hold on
scatter(x1(Rep),x2(Rep),20,'r')
scatter(x1(Ind),x2(Ind),20,'k')
title('111th US Senate in 2D')
```

- Projecting each senator’s votes in first two ***V*** coordinates, the right singular vectors: (1) is partisanship; (2) is bipartisanship
- Those coordinates are then plotted against each other
- Red: Republican
- Blue: Democrat
- Black: Independent
- The scatter plot suggests that there is a clear separation between parties!