

## AM 205: lecture 21

- ▶ New topic: eigenvalue problems
- ▶ Reminder: midterm this week
  - ▶ Posted online at 5 PM on Thursday 13th
  - ▶ Deadline at 5 PM on Friday 14th
  - ▶ Covers material up to and including lecture 16
  - ▶ Open book. No collaboration allowed.
  - ▶ Send questions as private messages on Piazza

# Motivation: Eigenvalue Problems

A matrix  $A \in \mathbb{C}^{n \times n}$  has eigenpairs  $(\lambda_1, v_1), \dots, (\lambda_n, v_n) \in \mathbb{C} \times \mathbb{C}^n$  such that

$$Av_i = \lambda v_i, \quad i = 1, 2, \dots, n$$

(We will order the eigenvalues from smallest to largest, so that  $|\lambda_1| \leq |\lambda_2| \leq \dots \leq |\lambda_n|$ )

It is remarkable how important eigenvalues and eigenvectors are in science and engineering!

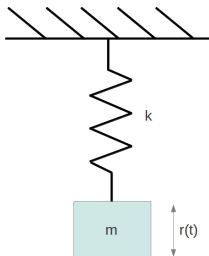
# Motivation: Eigenvalue Problems

For example, eigenvalue problems are closely related to **resonance**

- ▶ Pendulums
- ▶ Natural vibration modes of structures
- ▶ Musical instruments
- ▶ Lasers
- ▶ Nuclear Magnetic Resonance (NMR)
- ▶ ...

# Motivation: Resonance

Consider a spring connected to a mass



Suppose that:

- ▶ the spring satisfies Hooke's Law,<sup>1</sup>  $F(t) = ky(t)$
- ▶ a periodic forcing,  $r(t)$ , is applied to the mass

---

<sup>1</sup>Here  $y(t)$  denotes the position of the mass at time  $t$

## Motivation: Resonance

Then Newton's Second Law gives the ODE

$$y''(t) + \left(\frac{k}{m}\right) y(t) = r(t)$$

where  $r(t) = F_0 \cos(\omega t)$

Recall that the solution of this **non-homogeneous** ODE is the sum of a **homogeneous** solution,  $y_h(t)$ , and a **particular** solution,  $y_p(t)$

Let  $\omega_0 \equiv \sqrt{k/m}$ , then for  $\omega \neq \omega_0$  we get:<sup>2</sup>

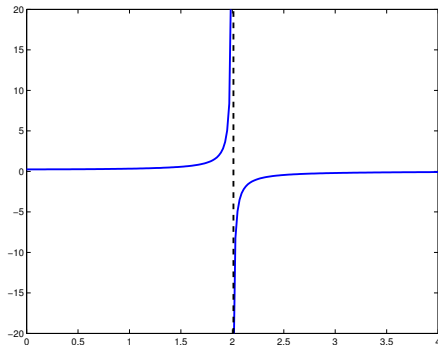
$$y(t) = y_h(t) + y_p(t) = C \cos(\omega_0 t - \delta) + \frac{F_0}{m(\omega_0^2 - \omega^2)} \cos(\omega t),$$

---

<sup>2</sup> $C$  and  $\delta$  are determined by the ODE initial condition

## Motivation: Resonance

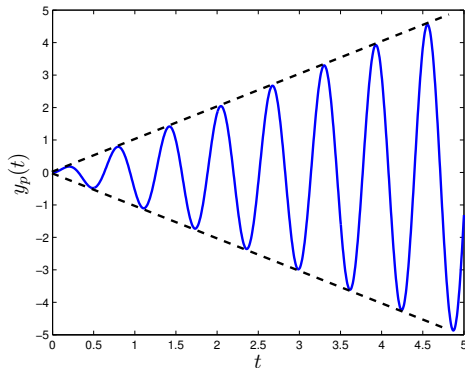
The amplitude of  $y_p(t)$ ,  $\frac{F_0}{m(\omega_0^2 - \omega^2)}$ , as a function of  $\omega$  is shown below



Clearly we observe singular behavior when the system is forced at its **natural frequency**, i.e. when  $\omega = \omega_0$

## Motivation: Forced Oscillations

Solving the ODE for  $\omega = \omega_0$  gives  $y_p(t) = \frac{F_0}{2m\omega_0} t \sin(\omega_0 t)$



This is **resonance!**

## Motivation: Resonance

In general,  $\omega_0$  is the frequency at which the **unforced system** has a non-zero oscillatory solution

For the single spring-mass system we substitute<sup>3</sup> the oscillatory solution  $y(t) \equiv x e^{i\omega_0 t}$  into  $y''(t) + \left(\frac{k}{m}\right) y(t) = 0$

This gives a scalar equation for  $\omega_0$ :

$$kx = \omega_0^2 mx \implies \omega_0 = \sqrt{k/m}$$

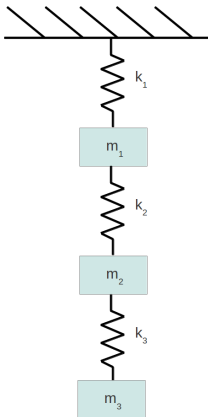
---

<sup>3</sup>Here  $x$  is the amplitude



## Motivation: Resonance

Suppose now we have a spring-mass system with three masses and three springs



## Motivation: Resonance

In the unforced case, this system is governed by the ODE system

$$My''(t) + Ky(t) = 0,$$

where  $M$  is the [mass matrix](#) and  $K$  is the [stiffness matrix](#)

$$M \equiv \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix}, \quad K \equiv \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix}$$

We again seek a nonzero oscillatory solution to this ODE, i.e. set  $y(t) = xe^{i\omega t}$ , where now  $y(t) \in \mathbb{R}^3$

This gives the algebraic equation

$$Kx = \omega^2 Mx$$

## Motivation: Eigenvalue Problems

Setting  $A \equiv M^{-1}K$  and  $\lambda = \omega^2$ , this gives the eigenvalue problem

$$Ax = \lambda x$$

Here  $A \in \mathbb{R}^{3 \times 3}$ , hence we obtain natural frequencies from the three eigenvalues  $\lambda_1, \lambda_2, \lambda_3$

## Motivation: Eigenvalue Problems

The spring-mass systems we have examined so far contain **discrete** components

But the same ideas also apply to **continuum models**

For example, the wave equation models vibration of a string (1D) or a drum (2D)

$$\frac{\partial^2 u(x, t)}{\partial t^2} - c \Delta u(x, t) = 0$$

As before, we write  $u(x, t) = \tilde{u}(x)e^{i\omega t}$ , to obtain

$$-\Delta \tilde{u}(x) = \frac{\omega^2}{c} \tilde{u}(x)$$

which is a **PDE eigenvalue problem**

# Motivation: Eigenvalue Problems

We can discretize the Laplacian operator with finite differences to obtain an algebraic eigenvalue problem

$$Av = \lambda v,$$

where

- ▶ the eigenvalue  $\lambda = \omega^2/c$  gives a natural vibration frequency of the system
- ▶ the eigenvector (or eigenmode)  $v$  gives the corresponding vibration mode

# Motivation: Eigenvalue Problems

We will use the Python (and Matlab) functions `eig` and `eigs` to solve eigenvalue problems

- ▶ `eig`: find all eigenvalues/eigenvectors of a **dense** matrix
- ▶ `eigs`: find a few eigenvalues/eigenvectors of a **sparse** matrix

We will examine the algorithms used by `eig` and `eigs` in this Unit

## Motivation: Eigenvalue Problems

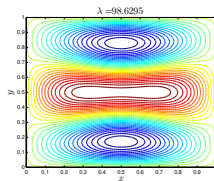
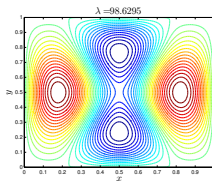
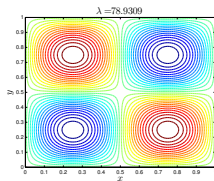
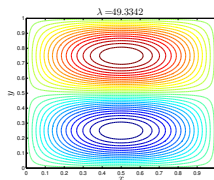
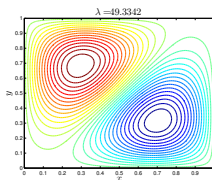
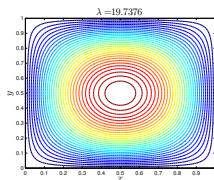
Python demo: Eigenvalues/eigenmodes of Laplacian on  $[0, 1]^2$ , zero Dirichlet boundary conditions

Based on separation of variables, we know that eigenmodes are  $\sin(\pi ix) \sin(\pi jy)$ ,  $i, j = 1, 2, \dots$

Hence eigenvalues are  $(i^2 + j^2)\pi^2$

$i$	$j$	$\lambda_{i,j}$
1	1	$2\pi^2 \approx 19.74$
1	2	$5\pi^2 \approx 49.35$
2	1	$5\pi^2 \approx 49.35$
2	2	$8\pi^2 \approx 78.96$
1	3	$10\pi^2 \approx 98.97$
$\vdots$	$\vdots$	$\vdots$

# Motivation: Eigenvalue Problems



In general, for repeated eigenvalues, computed eigenmodes are L.I. members of the corresponding eigenspace

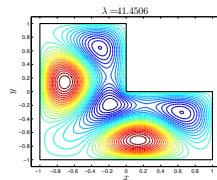
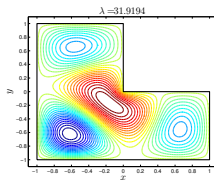
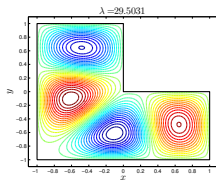
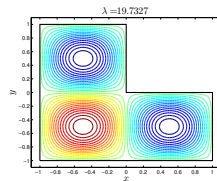
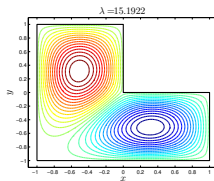
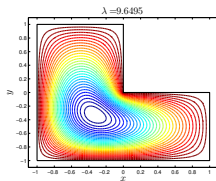
e.g. eigenmodes corresponding to  $\lambda = 49.3$  are given by

$$\alpha_{1,2} \sin(\pi x) \sin(\pi 2y) + \alpha_{2,1} \sin(\pi 2x) \sin(\pi y), \quad \alpha_{1,2}, \alpha_{2,1} \in \mathbb{R}$$



# Motivation: Eigenvalue Problems

And of course we can compute eigenmodes of other shapes...



# Motivation: Eigenvalue Problems

A well-known mathematical question was posed by Mark Kac in 1966: “Can one hear the shape of a drum?”

The eigenvalues of a shape in 2D correspond to the resonant frequencies that a drumhead of that shape would have

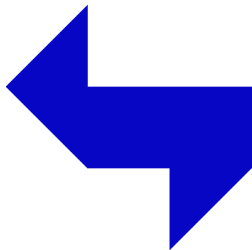
Therefore, the eigenvalues determine the harmonics, and hence the sound of the drum

So in mathematical terms, Kac's question was: If we know all of the eigenvalues, can we uniquely determine the shape?

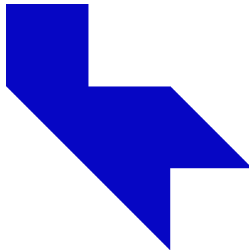
# Motivation: Eigenvalue Problems

It turns out that the answer is **no**!

In 1992, Gordon, Webb, and Wolpert constructed two different 2D shapes that have exactly the same eigenvalues!



Drum 1



Drum 2

## Motivation: Eigenvalue Problems

We can compute the eigenvalues and eigenmodes of the Laplacian on these two shapes using the algorithms from this Unit<sup>4</sup>

The first five eigenvalues are computed as:

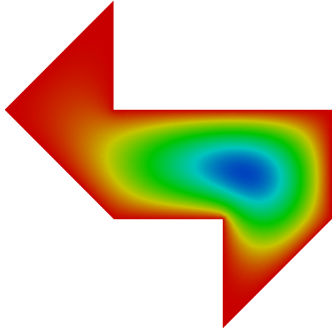
	Drum 1	Drum 2
$\lambda_1$	2.54	2.54
$\lambda_2$	3.66	3.66
$\lambda_3$	5.18	5.18
$\lambda_4$	6.54	6.54
$\lambda_5$	7.26	7.26

We next show the corresponding eigenmodes...

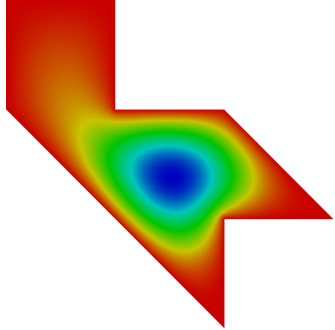
---

<sup>4</sup>Note here we employ the Finite Element Method (outside the scope of AM205), an alternative to F.D. that is well-suited to complicated domains

## Motivation: Eigenvalue Problems

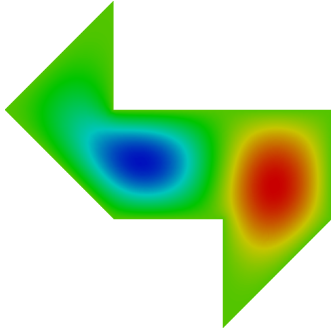


eigenmode 1

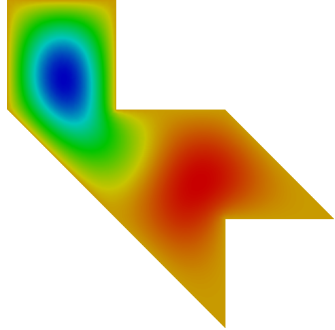


eigenmode 1

## Motivation: Eigenvalue Problems

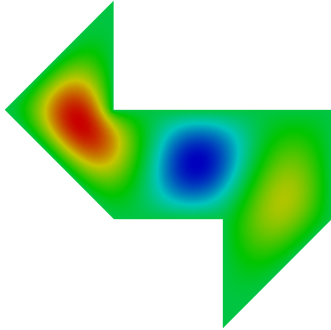


eigenmode 2

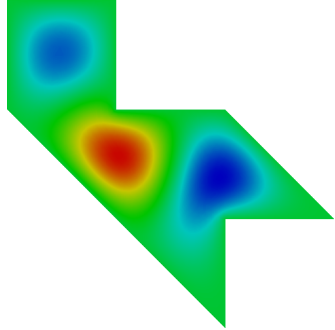


eigenmode 2

## Motivation: Eigenvalue Problems

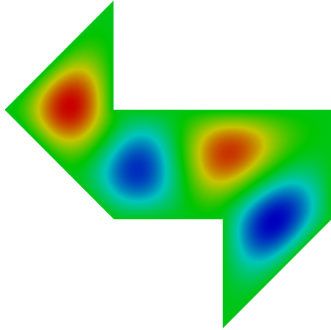


eigenmode 3

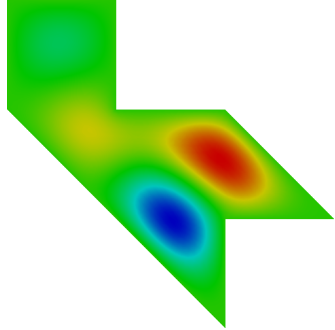


eigenmode 3

## Motivation: Eigenvalue Problems



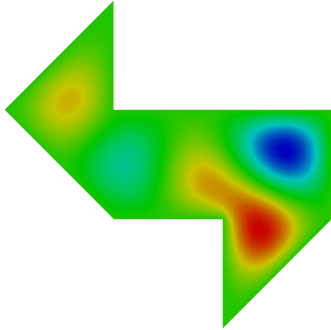
eigenmode 4



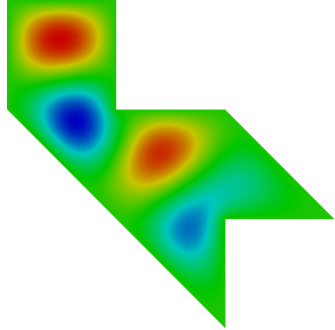
eigenmode 4



## Motivation: Eigenvalue Problems



eigenmode 5



eigenmode 5

# Eigenvalues and Eigenvectors

Eigenvalues and eigenvectors of real-valued matrices can be complex

Hence in this Unit we will generally work with **complex-valued matrices and vectors**,  $A \in \mathbb{C}^{n \times n}$ ,  $v \in \mathbb{C}^n$

For  $A \in \mathbb{C}^{n \times n}$ , we shall consider the eigenvalue problem: find  $(\lambda, v) \in \mathbb{C} \times \mathbb{C}^n$  such that

$$\begin{aligned}Av &= \lambda v, \\ \|v\|_2 &= 1\end{aligned}$$

Note that for  $v \in \mathbb{C}^n$ ,  $\|v\|_2 \equiv (\sum_{k=1}^n |v_k|^2)^{1/2}$ , where  $|\cdot|$  is the **modulus** of a complex number

# Eigenvalues and Eigenvectors

This problem can be reformulated as:

$$(A - \lambda I)v = 0$$

We know this system has a solution if and only if  $(A - \lambda I)$  is singular, hence we must have

$$\det(A - \lambda I) = 0$$

$p(z) \equiv \det(A - zI)$  is a degree  $n$  polynomial, called the **characteristic polynomial** of  $A$

The eigenvalues of  $A$  are exactly the roots of the characteristic polynomial

# Characteristic Polynomial

By the fundamental theorem of algebra, we can factorize  $p(z)$  as

$$p(z) = c_n(z - \lambda_1)(z - \lambda_2) \cdots (z - \lambda_n),$$

where the roots  $\lambda_i \in \mathbb{C}$  need not be distinct

Note also that complex eigenvalues of a matrix  $A \in \mathbb{R}^{n \times n}$  must occur as **complex conjugate pairs**

That is, if  $\lambda = \alpha + i\beta$  is an eigenvalue, then so is its complex conjugate  $\bar{\lambda} = \alpha - i\beta$

# Characteristic Polynomial

This follows from the fact that for a polynomial  $p$  with **real coefficients**,  $p(\bar{z}) = \overline{p(z)}$  for any  $z \in \mathbb{C}$ :

$$p(\bar{z}) = \sum_{k=0}^n c_k (\bar{z})^k = \sum_{k=0}^n c_k \overline{z^k} = \overline{\sum_{k=0}^n c_k z^k} = \overline{p(z)}$$

Hence if  $w \in \mathbb{C}$  is a root of  $p$ , then so is  $\bar{w}$ , since

$$0 = p(w) = \overline{p(w)} = p(\bar{w})$$

# Companion Matrix

We have seen that every matrix has an associated characteristic polynomial

Similarly, every polynomial has an associated **companion matrix**

The companion matrix,  $C_n$ , of  $p \in \mathbb{P}_n$  is a matrix which has eigenvalues that match the roots of  $p$

Divide  $p$  by its leading coefficient to get a **monic** polynomial, i.e. with leading coefficient equal to 1 (this doesn't change the roots)

$$p_{\text{monic}}(z) = c_0 + c_1z + \cdots + c_{n-1}z^{n-1} + z^n$$

## Companion Matrix

Then  $p_{\text{monic}}$  is the characteristic polynomial of the  $n \times n$  companion matrix

$$C_n = \begin{bmatrix} 0 & 0 & \cdots & 0 & -c_0 \\ 1 & 0 & \cdots & 0 & -c_1 \\ 0 & 1 & \cdots & 0 & -c_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -c_{n-1} \end{bmatrix}$$

# Companion Matrix

We show this for the  $n = 3$  case: Consider

$$p_{3,\text{monic}}(z) \equiv c_0 + c_1 z + c_2 z^2 + z^3,$$

which has companion matrix

$$C_3 \equiv \begin{bmatrix} 0 & 0 & -c_0 \\ 1 & 0 & -c_1 \\ 0 & 1 & -c_2 \end{bmatrix}$$

Recall that for a  $3 \times 3$  matrix, we have

$$\det \left( \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \right) = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} \\ - a_{13}a_{22}a_{31} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33}$$



## Companion Matrix

Substituting entries of  $C_3$  then gives

$$\det(zI - C_3) = c_0 + c_1z + c_2z^2 + z^3 = p_{3,\text{monic}}(z)$$

This link between matrices and polynomials is used by Python's `roots` function

`roots` computes all roots of a polynomial by using algorithms considered in this Unit to find eigenvalues of the companion matrix

# Eigenvalue Decomposition

Let  $\lambda$  be an eigenvalue of  $A \in \mathbb{C}^{n \times n}$ ; the set of all eigenvalues is called the **spectrum of  $A$**

The **algebraic multiplicity** of  $\lambda$  is the multiplicity of the corresponding root of the characteristic polynomial

The **geometric multiplicity** of  $\lambda$  is the number of linearly independent eigenvectors corresponding to  $\lambda$

For example, for  $A = I$ ,  $\lambda = 1$  is an eigenvalue with algebraic and geometric multiplicity of  $n$

(Char. poly. for  $A = I$  is  $p(z) = (z - 1)^n$ , and  $e_i \in \mathbb{C}^n$ ,  $i = 1, 2, \dots, n$  are eigenvectors)

# Eigenvalue Decomposition

**Theorem:** The geometric multiplicity of an eigenvalue is less than or equal to its algebraic multiplicity

If  $\lambda$  has geometric multiplicity  $<$  algebraic multiplicity, then  $\lambda$  is said to be **defective**

We say a matrix is defective if it has at least one defective eigenvalue

# Eigenvalue Decomposition

For example,

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{bmatrix}$$

has one eigenvalue with algebraic multiplicity of 3, geometric multiplicity of 1

```
Python 2.7.6 (default, Sep  9 2014, 15:04:36)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.39)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> a=np.array([[2,1,0],[0,2,1],[0,0,2]])
>>> v,d=np.linalg.eig(a)
>>> v
array([ 2.,  2.,  2.])
>>> d
array([[ 1.00000000e+00, -1.00000000e+00,  1.00000000e+00],
       [ 0.00000000e+00,  4.44089210e-16, -4.44089210e-16],
       [ 0.00000000e+00,  0.00000000e+00,  1.97215226e-31]])
```

# Eigenvalue Decomposition

Let  $A \in \mathbb{C}^{n \times n}$  be a **nondefective** matrix, then it has a full set of  $n$  linearly independent eigenvectors  $v_1, v_2, \dots, v_n \in \mathbb{C}^n$

Suppose  $V \in \mathbb{C}^{n \times n}$  contains the eigenvectors of  $A$  as columns, and let  $D = \text{diag}(\lambda_1, \dots, \lambda_n)$

Then  $Av_i = \lambda_i v_i, i = 1, 2, \dots, n$  is equivalent to  $AV = VD$

Since we assumed  $A$  is nondefective, we can invert  $V$  to obtain

$$A = VDV^{-1}$$

This is the **eigendecomposition** of  $A$

This shows that for a non-defective matrix,  $A$  is **diagonalized** by  $V$

# Eigenvalue Decomposition

We introduce the **conjugate transpose**,  $A^* \in \mathbb{C}^{n \times m}$ , of a matrix  $A \in \mathbb{C}^{m \times n}$

$$(A^*)_{ij} = \overline{A_{ji}}, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n$$

A matrix is said to be **hermitian** if  $A = A^*$  (this generalizes matrix symmetry)

A matrix is said to be **unitary** if  $AA^* = I$  (this generalizes the concept of an orthogonal matrix)

Also, for  $v \in \mathbb{C}^n$ ,  $\|v\|_2 = \sqrt{v^*v}$

# Eigenvalue Decomposition

In Python the `.T` operator performs the transpose, while the `.getH` operator performs the conjugate transpose

```
Python 2.7.6 (default, Sep  9 2014, 15:04:36)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.39)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> a=np.matrix([[1+1j,2+3j],[0,4]])
>>> a.T
matrix([[ 1.+1.j,  0.+0.j],
        [ 2.+3.j,  4.+0.j]])
>>> a.getH()
matrix([[ 1.-1.j,  0.-0.j],
        [ 2.-3.j,  4.-0.j]])
```

# Eigenvalue Decomposition

In some cases, the eigenvectors of  $A$  can be chosen such that they are orthonormal

$$v_i^* v_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

In such a case, the matrix of eigenvectors,  $Q$ , is unitary, and hence  $A$  can be **unitarily diagonalized**

$$A = QDQ^*$$



# Eigenvalue Decomposition

**Theorem:** A hermitian matrix is unitarily diagonalizable, and its eigenvalues are real

But hermitian matrices are not the only matrices that can be unitarily diagonalized...  $A \in \mathbb{C}^{n \times n}$  is **normal** if

$$A^*A = AA^*$$

**Theorem:** A matrix is unitarily diagonalizable if and only if it is normal

# Gershgorin's Theorem

Due to the link between eigenvalues and polynomial roots, in general one has to use iterative methods to compute eigenvalues

However, it is possible to gain some information about eigenvalue locations more easily from **Gershgorin's Theorem**

Let  $D(c, r) \equiv \{x \in \mathbb{C} : |x - c| \leq r\}$  denote a disk in the complex plane centered at  $c$  with radius  $r$

For a matrix  $A \in \mathbb{C}^{n \times n}$ ,  $D(a_{ii}, R_i)$  is called a **Gershgorin disk**, where

$$R_i = \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|,$$

# Gershgorin's Theorem

**Theorem:** All eigenvalues of  $A \in \mathbb{C}^{n \times n}$  are contained within the union of the  $n$  Gershgorin disks of  $A$

**Proof:** See lecture

# Gershgorin's Theorem

Note that a matrix is **diagonally dominant** if

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|, \quad \text{for } i = 1, 2, \dots, n$$

It follows from Gershgorin's Theorem that a diagonally dominant matrix cannot have a zero eigenvalue, hence **must be invertible**

For example, the finite difference discretization matrix of the differential operator  $-\Delta + \mathbf{I}$  is diagonally dominant

In 2-dimensions,  $(-\Delta + \mathbf{I})u = -u_{xx} - u_{yy} + u$

Each row of the corresponding discretization matrix contains diagonal entry  $4/h + 1$ , and four off-diagonal entries of  $-1/h$