

Oct. 4 2006

Lecture 9: Eigensystems of Matrix Equations

Reading:

Kreyszig Sections: 8.1, 8.2, 8.3 (pages 334–338, 340–343, 345–348)

Eigenvalues and Eigenvectors of a Matrix

The conditions for which general linear equation

$$\underline{A}\vec{x} = \vec{b} \quad (9-1)$$

has solutions for a given matrix \underline{A} , fixed vector \vec{b} , and unknown vector \vec{x} have been determined.

The operation of a matrix on a vector—whether as a physical process, or as a geometric transformation, or just a general linear equation—has also been discussed.

Eigenvalues and eigenvectors are among the most important mathematical concepts with a very large number of applications in physics and engineering.

An eigenvalue problem (associated with a matrix \underline{A}) relates the operation of a matrix multiplication on a particular vector \vec{x} to its multiplication by a particular scalar λ .

$$\underline{A}\vec{x} = \lambda\vec{x} \quad (9-2)$$

This equation bespeaks that the matrix operation can be replaced—or is equivalent to—a stretching or contraction of the vector: “ \underline{A} has some vector \vec{x} for which its multiplication is simply a scalar multiplication operation by λ .” \vec{x} is an *eigenvector* of \underline{A} and λ is \vec{x} 's associated *eigenvalue*.

The condition that Eq. 9-2 has solutions is that its associated homogeneous equation:

$$(\underline{A} - \lambda\underline{I})\vec{x} = \vec{0} \quad (9-3)$$

has a zero determinant:

$$\det(\underline{A} - \lambda\underline{I}) = 0 \quad (9-4)$$

Eq. 9-4 is a polynomial equation in λ (the power of the polynomial is the same as the size of the square matrix).

The eigenvalue-eigenvector system in Eq. 9-2 is solved by the following process:

1. Solve the characteristic equation (Eq. 9-4) for each of its roots λ_i .

2. Each root λ_i is used as an eigenvalue in Eq. 9-2 which is solved for its associated eigenvector \vec{x}_i

Lecture 09 MATHEMATICA® Example 1

Matrix eigensystems and their geometrical interpretation

Download notebooks, pdfs, or html from <http://pruffle.mit.edu/3.016-2006>.

The symbolic computation of eigenvalues and eigenvectors is demonstrated. In this example, a “cheat” is employed so that a matrix with “interesting” eigenvalues and eigenvectors is used as computation fodder. Just why the “cheat” works is demonstrated in Example 09-2.

1-3 These are initialization steps used to define `myMatrix` below. The idea here is to produce a non-trivial matrix with simple eigenvalues.

4: A “typical” 2×2 matrix A is defined...

5: Its eigenvalues are obtained by first solving equation Eq. 9-4 for λ ...

6: And, its eigenvectors could be obtained by putting each eigenvalue back into Eq. 9-2 and then solving \vec{x} for each distinct λ . However, this tedious procedure can also be performed with `Eigenvectors`

7: Here, a matrix of eigenvectors is defined with named rows `vec1` and `vec2`.

8: `Eigensystem` generates the same results as `Eigenvectors` and `Eigenvalues` in one step.

```

Initialization Steps: Define a 2x2 matrix and compute its
eigensystem
1 mtemp = DiagonalMatrix[{2 Pi, 4}];
  mtemp // MatrixForm
2 <<"Geometry' Rotations"
3 MatrixForm[msim = Transpose[RotationMatrix2D[Pi/4]].
  mtemp.RotationMatrix2D[Pi/4]]
4 mymatrix = {{2 + Pi, -2 + Pi}, {-2 + Pi, 2 + Pi}};
  mymatrix // MatrixForm
5 Solve[Det[mymatrix - λ IdentityMatrix[2]] == 0, λ]
6 Eigenvectors[mymatrix]
7 {vec1, vec2} = Eigenvectors[mymatrix]
8 Eigensystem[mymatrix]

```


For each position i , the j -terms can be added to each side, leaving an operator that looks like:

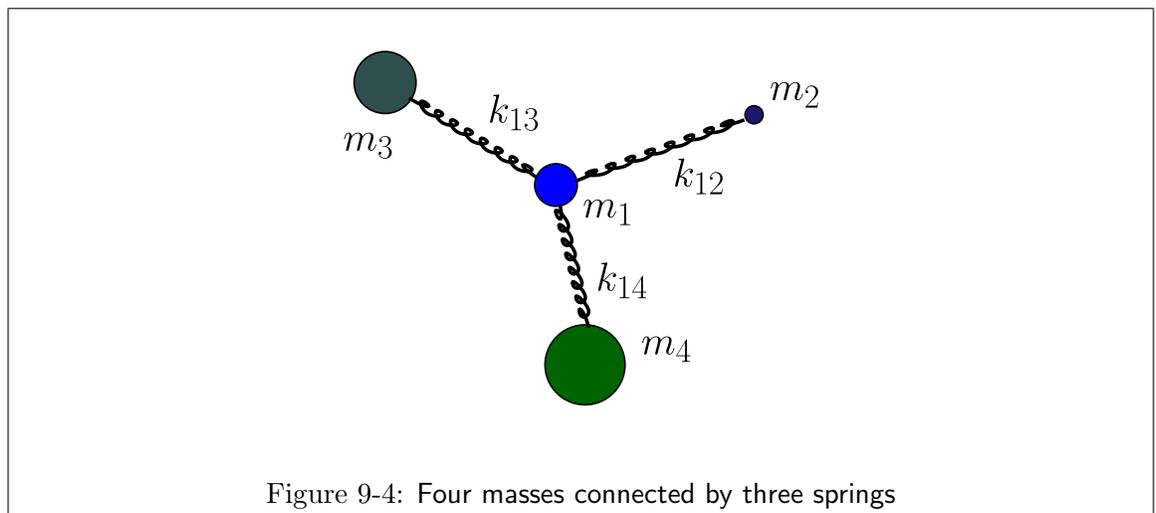
$$\mathcal{L}_{\text{lattice}} = \begin{pmatrix} m_1 \frac{d^2}{dt^2} & -k_{12} & 0 & -k_{14} & \dots & 0 \\ -k_{21} & m_2 \frac{d^2}{dt^2} & -k_{23} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & m_i \frac{d^2}{dt^2} & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & m_{N-1} \frac{d^2}{dt^2} & -k_{N-1 N} \\ 0 & 0 & \dots & -k_{N N-1} & m_N \frac{d^2}{dt^2} \end{pmatrix} \quad (9-8)$$

The operator $\mathcal{L}_{\text{lattice}}$ has diagonal entries that have the spring (second-derivative) operator and one off-diagonal entry for each other atom that interacts with the atom associated with row i . The system of atoms can be written as:

$$\underline{k}^{-1} \mathcal{L}_{\text{lattice}} \vec{x} = \vec{x} \quad (9-9)$$

which is another eigenvalue equation and solutions are constrained to have unit eigenvalues—these are the ‘normal modes.’

- To make the above example more concrete, consider a system of three masses connected by springs.



The equations of motion become:

$$\begin{pmatrix} m_1 \frac{d^2}{dt^2} & -k_{12} & -k_{13} & -k_{14} \\ -k_{12} & m_2 \frac{d^2}{dt^2} & 0 & 0 \\ -k_{13} & 0 & m_2 \frac{d^2}{dt^2} & 0 \\ -k_{14} & 0 & 0 & m_2 \frac{d^2}{dt^2} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} k_{12} + k_{13} + k_{14} & 0 & 0 & 0 \\ 0 & k_{12} & 0 & 0 \\ 0 & 0 & k_{13} & 0 \\ 0 & 0 & 0 & k_{14} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \quad (9-10)$$

which can be written as

$$\mathcal{L}_{4 \times 4} \vec{x} = \underline{k} \vec{x} \quad (9-11)$$

or

$$\underline{k}^{-1} \mathcal{L}_{4 \times 4} \vec{x} = \vec{x} \quad (9-12)$$

As will be discussed later, this system of equations can be “diagonalized” so that it becomes four independent equations. Diagonalization depends on finding the eigensystem for the operator.

- The one-dimensional Schrödinger wave equation is:

$$-\frac{\hbar}{2m} \frac{d^2\psi(x)}{dx^2} + U(x)\psi(x) = E\psi(x) \quad (9-13)$$

where the second derivative represents the kinetic energy and $U(x)$ is the spatial-dependent potential energy. The “Hamiltonian Operator” $\mathcal{H} = -\frac{\hbar}{2m} \frac{d^2}{dx^2} + U(x)$, operates on the wavefunction $\psi(x)$ and returns the wavefunction’s total energy multiplied by the wavefunction;

$$\mathcal{H}\psi(x) = E\psi(x) \quad (9-14)$$

This is another important eigenvalue equation (and concept!)

Symmetric, Skew-Symmetric, Orthogonal Matrices

Three types of matrices occur repeatedly in physical models and applications. They can be placed into three categories according to the conditions that are associated with their eigenvalues:

All real eigenvalues Symmetric matrices—those that have a “mirror-plane” along the northwest-southeast diagonal ($\underline{A} = \underline{A}^T$)—must have all real eigenvalues.

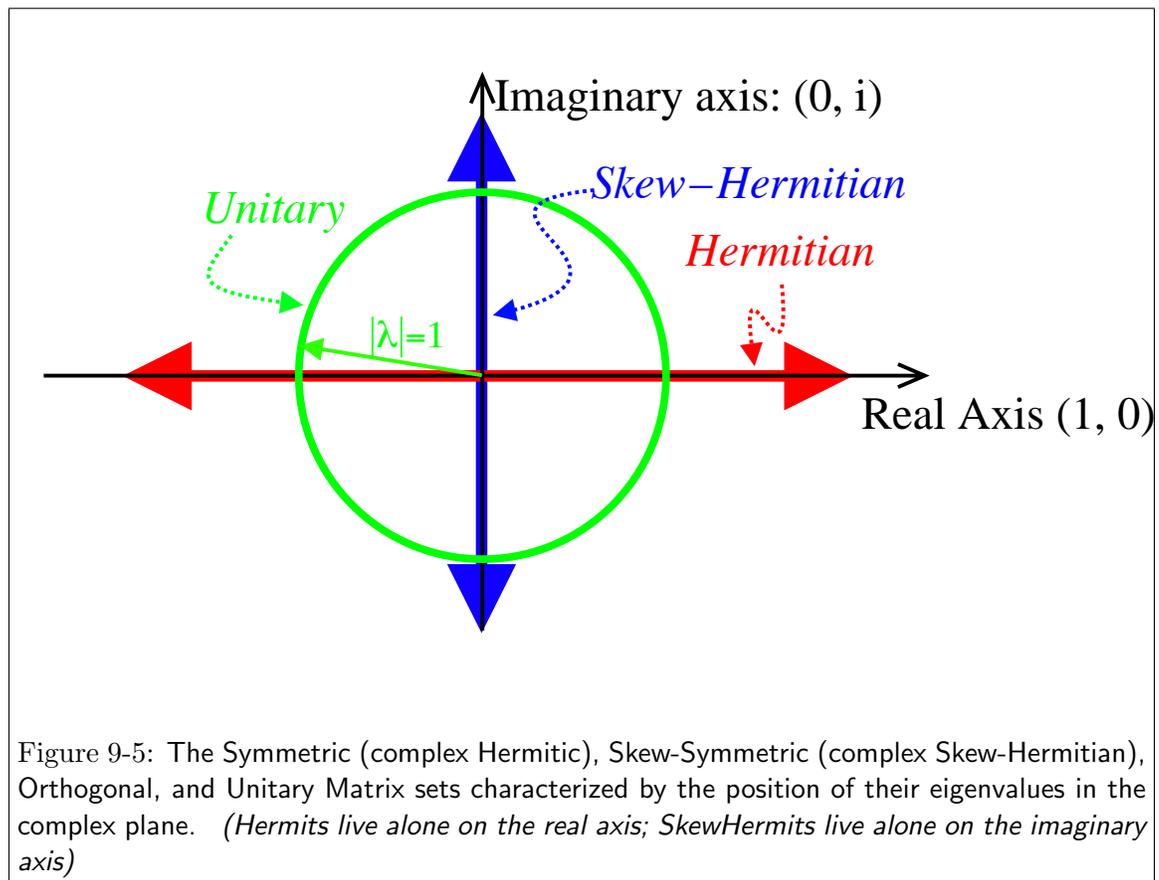
Hermitian matrices—the complex analogs of symmetric matrices—in which the reflection across the diagonal is combined with a complex conjugate operation ($a_{ij} = \bar{a}_{ji}$), must also have all real eigenvalues.

All imaginary eigenvalues Skew-symmetric (diagonal mirror symmetry combined with a minus) matrices ($-\underline{A} = \underline{A}^T$) must have all complex eigenvalues.

Skew-Hermitian matrices—the complex analogs of skew-symmetric matrices ($a_{ij} = -\bar{a}_{ji}$)—have all imaginary eigenvalues.

Unitary Matrices: unit determinant Real matrices that satisfy $\underline{A}^T = \underline{A}^{-1}$ have the property that product of all the eigenvalues is ± 1 . These are called *orthogonal* matrices and they have *orthonormal* rows. Their determinants are also ± 1 .

This is generalized by complex matrices that satisfy $\bar{\underline{A}}^T = \underline{A}^{-1}$. These are called *unitary* matrices and their (complex) determinants have magnitude 1. Orthogonal matrices, \underline{A} , have the important physical property that they preserve the inner product: $\vec{x} \cdot \vec{y} = (\underline{A}\vec{x}) \cdot (\underline{A}\vec{y})$. When the orthogonal matrix is a rotation, the interpretation is that the vectors maintain their relationship to each other if they are both rotated.



Orthogonal Transformations

Multiplication of a vector by an orthogonal matrix is equivalent to an orthogonal geometric transformation on that vector.

For orthogonal transformation, the inner product between any two vectors is *invariant*. That is, the inner product of two vectors is always the same as the inner product of their images under an orthogonal transformation. Geometrically, the projection (or the angular relationship) is unchanged. This is characteristic of a rotation, or a reflection, or an inversion.

Rotations, reflections, and inversions are orthogonal transformations. The product of orthogonal matrices is also an orthogonal matrix.

Lecture 09 MATHEMATICA® Example 2

Coordinate Transformations to The Eigenbasis

Download notebooks, pdfs, or html from <http://pruffle.mit.edu/3.016-2006>.

Demonstrates that the matrix composed of columns constructed eigenvectors of a matrix can be used to *diagonalize a matrix* and the resulting diagonal entries are the matrix eigenvalues.

- 1: `simtrans` is constructed by assigning rows defined by the eigenvectors from Example 09-1 and then transposing so that the eigenvectors are the columns.
- 2: The original matrix is left-multiplied by the *inverse* of `simtrans` and right-multiplied by `simtrans`; the result will be a diagonal matrix with the original matrix's eigenvalues as diagonal entries.
- 2-4 The eigenvectors are orthogonal. There is a process called *Gram-Schmidt* orthogonalization used to define a set of vectors that are normal to each other. Here, `GramSchmidt` produces vectors that are also *normalized to unit vectors*. This, and other useful vector functions such as `Normalize` are available in the `LinearAlgebra`Orthogonalization` package.
- 5-6 The geometrical interpretation of the matrix that performs the diagonalization can be obtained by loading the `Geometry`Rotations` package. The eigenvector-matrix can be compared to the $\pi/4$ rotation matrix.
- 8-9 Demonstrates that Eq. 9-2 is true.
- 10: Demonstrates that $\underline{A}^n \vec{x} = \lambda^n \vec{x}$.

```

1 simtrans = {evec2, evec1} // Transpose;
  simtrans // MatrixForm
2 Inverse[simtrans].mymatrix.simtrans // Simplify // MatrixForm
3 << LinearAlgebra`Orthogonalization`
4 GramSchmidt[Eigenvectors[mymatrix]] // MatrixForm
5 << Geometry`Rotations`
6 MatrixForm[RotationMatrix2D[π/4]]
7 evec1
  evec2
8 mymatrix.evec1
9 mymatrix.evec2
10 MatrixPower[mymatrix, 12].evec2 // Simplify

```