None

In [1]:

```
import sympy
from sympy import Matrix, Rational, sqrt, symbols, zeros, simplify
import numpy as np
import matplotlib.pyplot as plt
import ipywidgets as widgets
from ipywidgets import interact, interactive, fixed, interact_manual
import matplotlib.pyplot as plt
%matplotlib notebook
```

Mathematics for Machine Learning

Session 8: The determinant; eigenvectors and eigenvalues

Gerhard Jäger

November 14, 2024

Determinants

Sympy

In [2]:

```
a,b,c,d, e, f, g, h, i = symbols('a b c d e f g h i')
A = Matrix([
    [a, b],
    [c,d]
])
A
```

Out[2]:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

In [3]:

```
A.det()
```

Out[3]:

$$ad - bc$$

In [4]:

```
A = Matrix([
    [a,b,c],
    [d,e,f],
    [g,h,i]
])
A
```

Out[4]:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

In [5]:

```
l, u, _ = A.LUdecomposition()
```

In [6]:

```
sympy.simplify(u)
```

Out[6]:

$$\begin{bmatrix} a & b & c \\ 0 & e - \frac{bd}{a} & f - \frac{cd}{a} \\ 0 & 0 & \frac{aei - afh - bdi + bfg + cdh - ceg}{ae - bd} \end{bmatrix}$$

In [7]:

```
A
```

Out[7]:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

In [8]:

```
from math import prod
sympy.simplify(prod(u.diagonal()))
```

Out[8]:

$$aei - afh - bdi + bfg + cdh - ceg$$

In [9]:

```
A.det()
```

Out[9]:

$$aei - afh - bdi + bfg + cdh - ceg$$

Geometric interpretation

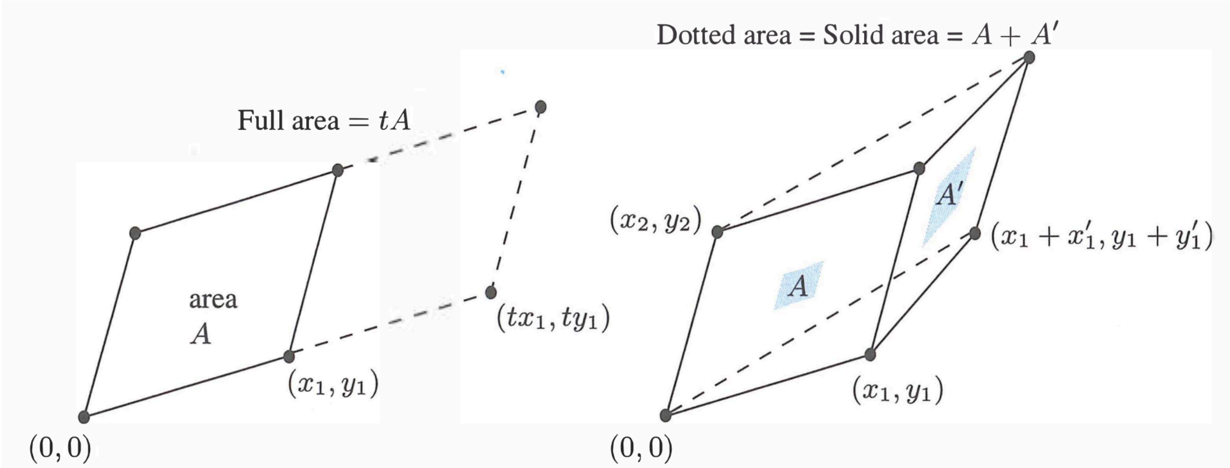
The absolute value A is the n-dimensional volume of the parallelepiped created by the column vectors of A.

Why?

It is fairly easy to see that this interpretation holds for the first two axioms:

- the parallelepiped corresponding to the identity matrix is the n-dimensional standard cube with length 1 along each edge
- swapping two columns does not change the parallelepiped

The third axiom is more complex. These pictures give an intuitive explanation in 2 dimensions:



There is no simple geometric interpretation of the sign of the determinant though.

Determinants and permutations

Consider our standard 2×2 matrix

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

According to rule 3:

$$\begin{aligned} \begin{vmatrix} a & b \\ c & d \end{vmatrix} &= \begin{vmatrix} a & b \\ 0 & d \end{vmatrix} + \begin{vmatrix} 0 & b \\ c & d \end{vmatrix} \\ &= \begin{vmatrix} a & b \\ 0 & 0 \end{vmatrix} + \begin{vmatrix} a & 0 \\ 0 & d \end{vmatrix} + \begin{vmatrix} 0 & b \\ c & 0 \end{vmatrix} + \begin{vmatrix} 0 & 0 \\ c & d \end{vmatrix} \end{aligned}$$

The first and last summand each have a zero row. According to rules 6 and 10, their determinants are 0. So we get

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = \begin{vmatrix} a & 0 \\ 0 & d \end{vmatrix} + \begin{vmatrix} 0 & b \\ c & 0 \end{vmatrix}$$

According to rule 3, it follows that

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} + bc \begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix}$$

Note the pattern:

- each summand is the determinant of a **permutation matrix**,
- multiplied by the entries of the original matrix corresponding to the non-zero entries of the permutation matrix

(Reminder: a permutation matrix is a square matrix with exactly one 1 in each row and each column, and 0 everywhere else.)

According to rules 2 and 1:

$$
\begin{aligned}
\begin{vmatrix} a & b \\ c & d \end{vmatrix} &= ad\begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} + bc\begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix} \\
&= ad\begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} - bc\begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} \\
&= ad - bc
\end{aligned}
$$

Same thing for 3×3 matrix:

$$
\begin{aligned}
\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} &= \begin{vmatrix} a & & \\ & e & \\ & & i \end{vmatrix} + \begin{vmatrix} a & & \\ & & f \\ & h & \end{vmatrix} + \begin{vmatrix} & b & \\ d & & \\ & & i \end{vmatrix} + \begin{vmatrix} & b & \\ & & f \\ g & & \end{vmatrix} + \begin{vmatrix} & & c \\ d & & \\ & h & \end{vmatrix} + \begin{vmatrix} & & c \\ & e & \\ g & & \end{vmatrix} \\
&= \begin{vmatrix} a & & \\ & e & \\ & & i \end{vmatrix} - \begin{vmatrix} a & & \\ & f & \\ & & h \end{vmatrix} - \begin{vmatrix} b & & \\ & d & \\ & & i \end{vmatrix} - \begin{vmatrix} b & & \\ & & f \\ & g & \end{vmatrix} - \begin{vmatrix} & c & \\ d & & \\ & & h \end{vmatrix} - \begin{vmatrix} & c & \\ & & e \\ g & & \end{vmatrix} \\
&= \begin{vmatrix} a & & \\ & e & \\ & & i \end{vmatrix} - \begin{vmatrix} a & & \\ & f & \\ & & h \end{vmatrix} - \begin{vmatrix} b & & \\ & d & \\ & & i \end{vmatrix} + \begin{vmatrix} b & & \\ & f & \\ & & g \end{vmatrix} + \begin{vmatrix} c & & \\ & d & \\ & & h \end{vmatrix} + \begin{vmatrix} c & & \\ & & e \\ & g & \end{vmatrix} \\
&= \begin{vmatrix} a & & \\ & e & \\ & & i \end{vmatrix} - \begin{vmatrix} a & & \\ & f & \\ & & h \end{vmatrix} - \begin{vmatrix} b & & \\ & d & \\ & & i \end{vmatrix} + \begin{vmatrix} b & & \\ & f & \\ & & g \end{vmatrix} + \begin{vmatrix} c & & \\ & d & \\ & & h \end{vmatrix} - \begin{vmatrix} c & & \\ & e & \\ & & g \end{vmatrix} \\
&= aei + bfg + cdh - ceg - bdi - afh
\end{aligned}
$$

Let π be a *permutation* of $1, \ldots, n$. This means π is a **bijection** from $\{1, \ldots, n\}$ onto itself.

Each component of the formula above has the form

$$\pm \prod_i a_{i,\pi(i)}$$

for some permutation π.

We distinguish *even* and *odd* permutations::

Definition

A permutation π is **even ***if and only if***

$$\|\{\langle i,j\rangle | i < j \wedge \pi(i) > \pi(j)\}\|$$

is even. Otherwise it is* odd.*

The sum in the definition is the number of column permutations we have to perform to convert the corresponding permutation matrix to the identity matrix.

Definition

Let π be a permutation.

$$\mathrm{sign}(\pi) \doteq \begin{cases} 1 \text{ if } \pi \text{ is even} \\ -1 \text{ else} \end{cases}$$

This leads to the **Leibniz formula**:

$$|A| = \sum_{\pi : \pi \text{ a permutation over } \{1,...,n\}} \mathrm{sign}(\pi) \prod_i a_{i,\pi(i)}$$

This formula is much too unwieldy for actual computations, but it is useful for proving properties of the determinant.

Cofactors

If we expand the Leibniz formula for $n = 3$, we get

$$\begin{aligned} |A| = & \quad a_{11}a_{22}a_{33} - a_{11}a_{23}a_{32} \\ & -a_{21}a_{12}a_{33} + a_{21}a_{32}a_{13} \\ & +a_{31}a_{12}a_{23} - a_{31}a_{22}a_{13} \\ = & \quad a_{11}(a_{22}a_{33} - a_{23}a_{32}) \\ & -a_{21}(a_{12}a_{33} - a_{32}a_{13}) \\ & +a_{31}(a_{12}a_{23} - a_{22}a_{13}) \end{aligned}$$

Note that in the second equation, we have three products. Each consists of

- $(-1)^{1+i}$
- an entry a_{i1} from the first column, and
- the determinant of the matrix that results when we remove from A the first column and the ith row, and

This generalizes to matrices of arbitrary size, and to arbitrary columns.

Let M_{ij} be the matrix that results if we remove the ith row and the jth column from A.

$$|A| = \sum_i (-1)^{i+j} a_{ij} |M_{ij}|$$

Fore brevity's sake, we define

$$C_{ij} \doteq (-1)^{i+j} |M_{ij}|$$

These quantities are called **cofactors**.

Then the above formula simplifies to the **Laplace expansion**

$$|A| = \sum_i a_{ij} C_{ij}$$

Note that all matrices M_{ij} have size $(n-1) \times (n-1)$.

This leads to a **recursive definition** of the determinant:

- *If* $n = 1$, $|A| = a_{11}$.
- *If* $n > 1$, $|A| = \sum_i (-1)^{i+1} a_{i1} |M_{i1}|$.

Applying this definition to actual computations is not advisable, because it amounts to an application of the Leibniz formula, i.e., it is computationally costly.

Matrix powers and exponentials

So far we studied basic operations such as addition and multiplication of matrices and vectors.

Higher operations are also defined, at least for square matrices:

power

- $A^0 = \mathbf{I}$
- $A^{n+1} = AA^n$

exponential

- $e^A = \sum_{k=0}^{\infty} \frac{1}{k!} A^k$

To compute them efficiently (and for many other applications), we need **eigenvectors** and **eigenvalues** of square matrices.

Example

In [10]:
```
A = Matrix([
    [0.8, 0.3],
    [0.2, 0.7]
])
A
```

Out[10]:

$$\begin{bmatrix} 0.8 & 0.3 \\ 0.2 & 0.7 \end{bmatrix}$$

In [11]:
```
A**2
```

Out[11]:

$$\begin{bmatrix} 0.7 & 0.45 \\ 0.3 & 0.55 \end{bmatrix}$$

In [12]:
```
A**3
```

Out[12]:

$$\begin{bmatrix} 0.65 & 0.525 \\ 0.35 & 0.475 \end{bmatrix}$$

In [13]:

```
A**10
```

Out[13]:

$$\begin{bmatrix} 0.600390625 & 0.5994140625 \\ 0.399609375 & 0.4005859375 \end{bmatrix}$$

In [14]:

```
A**100
```

Out[14]:

$$\begin{bmatrix} 0.600000000000002 & 0.600000000000002 \\ 0.400000000000001 & 0.400000000000001 \end{bmatrix}$$

In [15]:

```
A**1000000
```

Out[15]:

$$\begin{bmatrix} 0.6 & 0.6 \\ 0.4 & 0.4 \end{bmatrix}$$

Eigenvectors and eigenvalues

Basic equation

$$A\mathbf{x} = \lambda\mathbf{x}$$

- $\mathbf{x}$ is called an **eigenvector** of A
- λ is called an **eigenvalue** of A

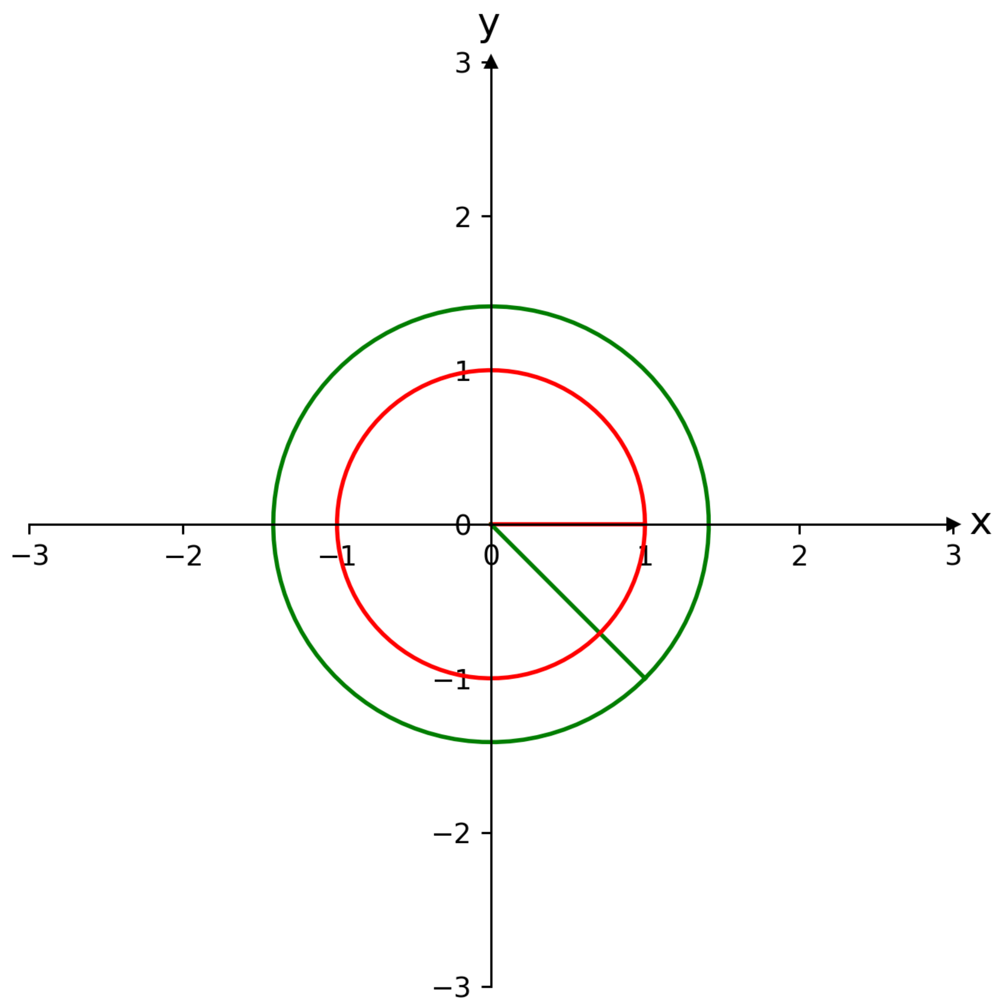
In [16]:

```
A = np.array([
    [1, 1],
    [-1, 1]
])
```

In [17]:

```
def g(A, theta):
    a, c = np.cos(theta), np.sin(theta)
    x = np.linspace(0, a, 100)
    y = np.linspace(0, c, 100)
    b, d = A @ np.array([a,c])
    z = np.linspace(0, b, 100)
    w = np.linspace(0, d, 100)
    return x, y, z, w

fig, ax = plt.subplots(figsize=(6,6))
xmin, xmax, ymin, ymax = -2, 2, -2, 2
ax.set(xlim=(xmin-1, xmax+1), ylim=(ymin-1, ymax+1), aspect='equal')
ax.spines['bottom'].set_position('zero')
```

```python
ax.spines['left'].set_position('zero')
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)

ax.set_xlabel('x', size=14, labelpad=-24, x=1.03)
ax.set_ylabel('y', size=14, labelpad=-21, y=1.02, rotation=0)
arrow_fmt = dict(markersize=4, color='black', clip_on=False)
ax.plot((1), (0), marker='>', transform=ax.get_yaxis_transform(), **arrow_fmt)
ax.plot((0), (1), marker='^', transform=ax.get_xaxis_transform(), **arrow_fmt)

x, y, z, w = g(A, 0)
line1, = ax.plot(x,y, color='red')
line2, = ax.plot(z,w, color='green')

angles = np.linspace(0, 2*np.pi, 100)
crc = np.array([np.cos(angles), np.sin(angles)])
elps = A @ crc
ax.plot(crc[0,:], crc[1,:], color='red')
ax.plot(elps[0,:], elps[1,:], color='green')

def update(theta = 0):
    x, y, z, w = g(A,theta)
    print("x = "+str(x[-1])+","+str(y[-1]))
    print("y = "+str(z[-1])+","+str(w[-1]))
    line1.set_data(x, y)
    line2.set_data(z, w)
    fig.canvas.draw_idle()

interact(update, theta = (0, 2*np.pi, 0.01));
```




```
interactive(children=(FloatSlider(value=0.0, description='theta', max=6.283185307179586, s
tep=0.01), Output())…
```

Example: How to find eigenvalues and eigenvectors

$$A = \begin{bmatrix} 0.8 & 0.3 \\ 0.2 & 0.7 \end{bmatrix}$$

$$\begin{aligned} A\mathbf{x} &= \lambda \mathbf{x} \\ &= \lambda \mathbf{I} \mathbf{x} \\ (A - \lambda \mathbf{I})\mathbf{x} &= \mathbf{0} \end{aligned}$$

The important matrix now is

$$A - \lambda \mathbf{I} = \begin{bmatrix} 0.8 - \lambda & 0.3 \\ 0.2 & 0.7 - \lambda \end{bmatrix}$$

We are looking for a value of λ such that

$$(A - \lambda \mathbf{I})\mathbf{x} = \mathbf{0}$$

with $\mathbf{x} \neq \mathbf{0}$ (otherwise the equation would be trivial).

It follows that $(A - \lambda\mathbf{I})$ is not invertible. Hence:

$$|(A - \lambda\mathbf{I})| = 0$$

Using the formula for the determinant of a 2×2 matrix:

$$(0.8 - \lambda)(0.7 - \lambda) - 0.2 \times 0.3 = 0$$

Simplifying:

$$\begin{aligned}(0.8 - \lambda)(0.7 - \lambda) - 0.2 \times 0.3 = 0 \\ \lambda^2 - 1.5\lambda + 0.56 - 0.06 = 0 \\ \lambda^2 - 1.5\lambda + 0.5 = 0\end{aligned}$$

There is a formula for finding the solution of quadratic equations (https://en.wikipedia.org/wiki/Quadratic_equation):

$$\begin{aligned}\lambda_{1/2} = \frac{3}{4} \pm \sqrt{(\frac{3}{4})^2 - \frac{1}{2}} \\ &= \frac{3}{4} \pm \sqrt{\frac{9-8}{16}} \\ &= \frac{3}{4} \pm \sqrt{\frac{1}{16}} \\ &= \frac{3}{4} \pm \frac{1}{4} \\ \lambda_1 &= 1 \\ \lambda_2 &= 0.5\end{aligned}$$

λ_1 and λ_2 are the eigenvalues of A. Now let's find the corresponding eigenvectors.

This amount to finding the nullspace of $A - \lambda\mathbf{I}$:

- λ_1

$$\begin{aligned}(A - \lambda_1\mathbf{I})\mathbf{x}_1 &= \mathbf{0} \\ \begin{bmatrix} -0.2 & 0.3 \\ 0.2 & -0.3 \end{bmatrix}\mathbf{x}_1 &= \mathbf{0} \\ \begin{bmatrix} -0.2 & 0.3 \\ 0 & 0 \end{bmatrix}\mathbf{x}_1 &= \mathbf{0} \\ \begin{bmatrix} 1 & -1.5 \\ 0 & 0 \end{bmatrix}\mathbf{x}_1 &= \mathbf{0} \\ \mathbf{x}_1 &= \begin{bmatrix} 1.5 \\ 1 \end{bmatrix}\end{aligned}$$

- λ_2

$$
\begin{aligned}
(A - \lambda_2\mathbf{I})\mathbf{x}_1 &= \mathbf{0} \\
\begin{bmatrix} 0.3 & 0.3 \\ 0.2 & 0.2 \end{bmatrix} \mathbf{x}_2 &= \mathbf{0} \\
\begin{bmatrix} 0.3 & 0.3 \\ 0 & 0 \end{bmatrix} \mathbf{x}_2 &= \mathbf{0} \\
\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}_2 &= \mathbf{0} \\
\mathbf{x}_2 &= \begin{bmatrix} -1 \\ 1 \end{bmatrix}
\end{aligned}
$$

$\mathbf{x}_1$ is the eigenvector *corresponding to* λ_1.

$\mathbf{x}_2$ is the eigenvector *corresponding to* λ_2.

Any non-zero multiples of $\mathbf{x}_1, \mathbf{x}_2$ are also eigenvectors. It is common practice to use normalized eigenvectors, i.e.\ eigenvectors with length 1.

$$
\begin{aligned}
\mathbf{v}_1 &= \frac{\mathbf{x}_1}{||\mathbf{x}_1||} \\
&= \frac{1}{\sqrt{13}} \begin{bmatrix} 3 \\ 2 \end{bmatrix} \\
\mathbf{v}_2 &= \frac{\mathbf{x}_1}{||\mathbf{x}_1||} \\
&= \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix}
\end{aligned}
$$

In [18]:

```
A = Matrix([
    [Rational(4,5), Rational(3,10)],
    [Rational(1,5), Rational(7,10)]
])
A
```

Out[18]:

$$\begin{bmatrix} \frac{4}{5} & \frac{3}{10} \\ \frac{1}{5} & \frac{7}{10} \end{bmatrix}$$

In [19]:

```
e1, e2 = A.eigenvects()
```

In [20]:

```
lambda1, _, v1 = e1
```

In [21]:

```
lambda1
```

Out[21]:

$$\frac{1}{2}$$

In [22]:
```
v1[0].normalized()
```

Out[22]:

$$\begin{bmatrix} -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix}$$

In [23]:
```
lambda2, _, v2 = e2
```

In [24]:
```
lambda2
```

Out[24]:

1

In [25]:
```
v2[0].normalized()
```

Out[25]:

$$\begin{bmatrix} \frac{3\sqrt{13}}{13} \\ \frac{2\sqrt{13}}{13} \end{bmatrix}$$

In [26]:
```
A = np.array([
    [0.8, 0.3],
    [0.2, 0.7]
])
A
```

Out[26]:
```
array([[0.8, 0.3],
       [0.2, 0.7]])
```

In [27]:
```
np.linalg.eig(A)
```

Out[27]:
```
EigResult(eigenvalues=array([1. , 0.5]), eigenvectors=array([[ 0.83205029, -0.70710678],
       [ 0.5547002 ,  0.70710678]]))
```

Procedure to find eigenvalues and eigenvectors:

1. Construct $A - \lambda\mathbf{I}$ with λ as unknown.
2. Set $|A - \lambda\mathbf{I}| = 0$ and solve for λ. All solutions are eigenvalues.
3. For each solution for λ, find the nullspace of $|A - \lambda\mathbf{I}| = 0$. Each vector in the nullspace is an eigenvector corresponding to this solution.

Example 2: Projection matrix

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

- find eigenvalues

$$\begin{vmatrix} \frac{1}{2}-\lambda & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2}-\lambda \end{vmatrix} = 0$$
$$(\frac{1}{2}-\lambda)^2 - \frac{1}{4} = 0$$
$$\lambda^2 - \lambda = 0$$
$$\lambda(\lambda - 1) = 0$$
$$\lambda_1 = 1$$
$$\lambda_2 = 0$$

- find eigenvectors
  - λ_1:

$$\begin{bmatrix} -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix} \mathbf{x}_1 = \mathbf{0}$$
$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

  - λ_2:

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \mathbf{x}_2 = \mathbf{0}$$
$$\mathbf{x}_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Example 3: Reflection matrix

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

- find eigenvalues

$$\begin{vmatrix} -\lambda & 1 \\ 1 & -\lambda \end{vmatrix} = 0$$
$$\lambda^2 - 1 = 0$$
$$\lambda^2 = 1$$
$$\lambda_1 = 1$$
$$\lambda_2 = -1$$

- find eigenvectors
  - λ_1:

$$\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \mathbf{x}_1 = \mathbf{0}$$

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

- λ_2:

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \mathbf{x}_2 = \mathbf{0}$$

$$\mathbf{x}_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$