Prepared by Julian García, based on material by
Arun Konagurthu

Workshop 5
Eigenvalues and Eigenvectors

FIT 3139

Computational Modelling and Simulation

MONASH University
Information Technology

Eigenvalues and eigenvectors

Find a non-zero vector $\vec{x}$ such that:

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

direction **magnitude**

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

$$A\vec{x} - \lambda\vec{x} = \vec{0}$$

$$\boxed{(A - \lambda I)}\,\vec{x} = \vec{0}$$

Must be singular

To hold for a non-trivial vector $\vec{x}$

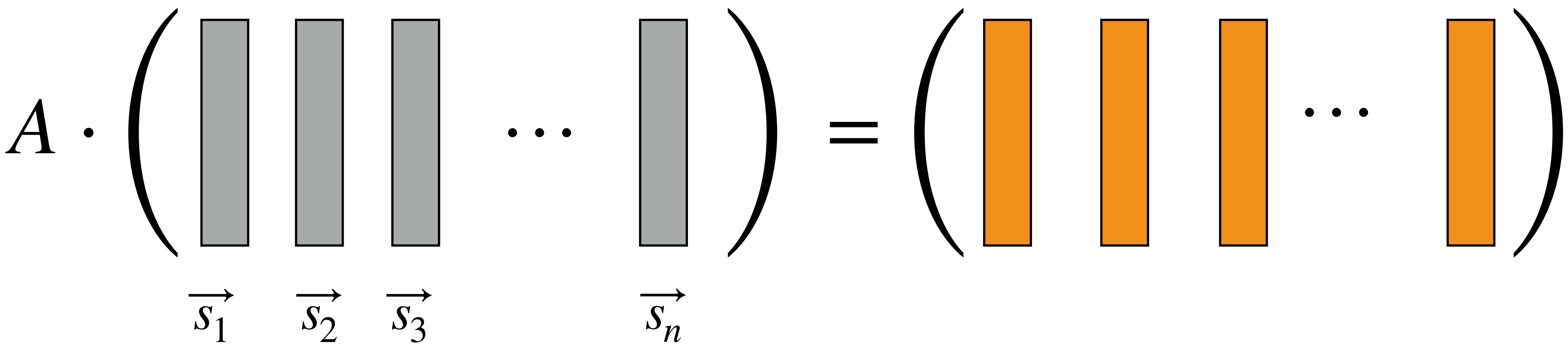
$$\boxed{|A - \lambda I| = 0}$$

Polynomial involving λ

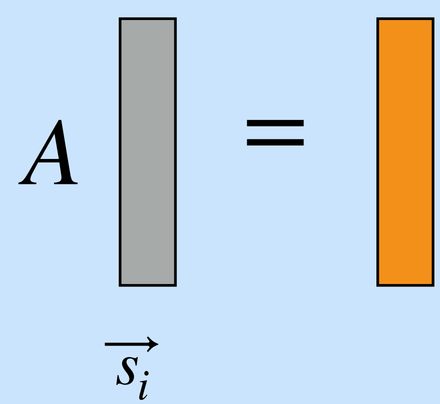
"characteristic polynomial" of degree n

$$
\begin{tikzcd}
\text{Matrix} \arrow[r] & \begin{array}{c}\text{Characteristic}\\ \text{Polynomial}\end{array} \arrow[r] & \text{Eigenvalues} \arrow[d] \\
& & \text{Direction}
\end{tikzcd}
$$

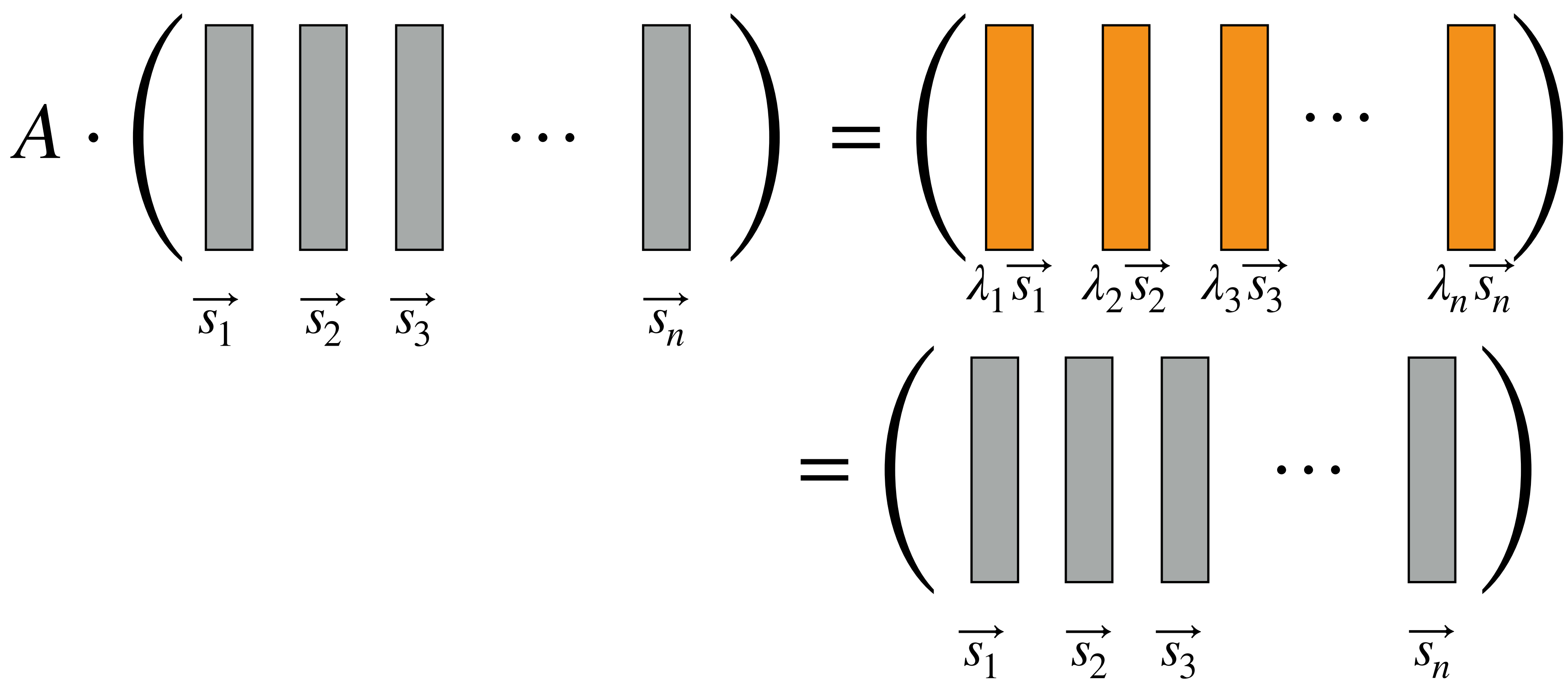
$$A_{nxn} \longrightarrow \begin{array}{ll} \{\overrightarrow{s_1}, \overrightarrow{s_2}, \overrightarrow{s_3}, \ldots, \overrightarrow{s_n}\} & \text{eigenvectors} \\ \{\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_n\} & \text{eigenvalues} \end{array}$$



$$A\overrightarrow{s_i} = \lambda_i \overrightarrow{s_i}$$



$$A_{nxn} \longrightarrow \begin{array}{c} \{\vec{s_1}, \vec{s_2}, \vec{s_3}, \ldots, \vec{s_n}\} \\ \{\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_n\} \end{array}$$



$$\begin{pmatrix} \lambda_1 & 0 & 0 & \ldots & 0 \\ 0 & \lambda_2 & 0 & \ldots & 0 \\ 0 & 0 & \lambda_3 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & \lambda_n \end{pmatrix}$$

$$AS = S\Lambda$$

$$ASS^{-1} = S\Lambda S^{-1}$$

$$A = S\Lambda S^{-1}$$

A expressed as a factorisation of a **matrix of eigenvectors** and a **diagonal matrix of eigenvalues**.

Eigen decomposition

Provides a **factorisation** of a given matrix in terms of the matrix's eigenvalues and eigenvectors

Eigen decomposition theorem:

Any $n \times n$ **diagonalisable*** matrix A can be factorised as:

$$A = S \Lambda S^{-1}$$

where:

- S is an $n \times n$ matrix of **eigenvectors** (as its columns), and
- Λ is a corresponding diagonal matrix of **eigenvalues** (as its diagonal elements)

A is called diagonalisable there exists an invertible matrix ***P*** and a diagonal matrix ***D*** such that $P^{-1}AP=D$*. Please see section 4.2.2 of the Heath book for an in-depth discussion.

matrix exponentiation

$$A^2 = (A)(A) = (S\Lambda S^{-1})(S\Lambda S^{-1}) = S\Lambda(\overbrace{\cancel{S^{-1}S}}^{\mathbf{I}})\Lambda S^{-1} = S\Lambda^2 S^{-1}$$

$$A^3 = (A^2)(A) = (S\Lambda^2 S^{-1})(S\Lambda S^{-1}) = S\Lambda^2(\overbrace{\cancel{S^{-1}S}}^{\mathbf{I}})\Lambda S^{-1} = S\Lambda^3 S^{-1}$$

$$\vdots$$

$$A^n = (A^{n-1})(A) = (S\Lambda^{n-1} S^{-1})(S\Lambda S^{-1}) = S\Lambda^{n-1}(\overbrace{\cancel{S^{-1}S}}^{\mathbf{I}})\Lambda S^{-1} = S\Lambda^n S^{-1}$$

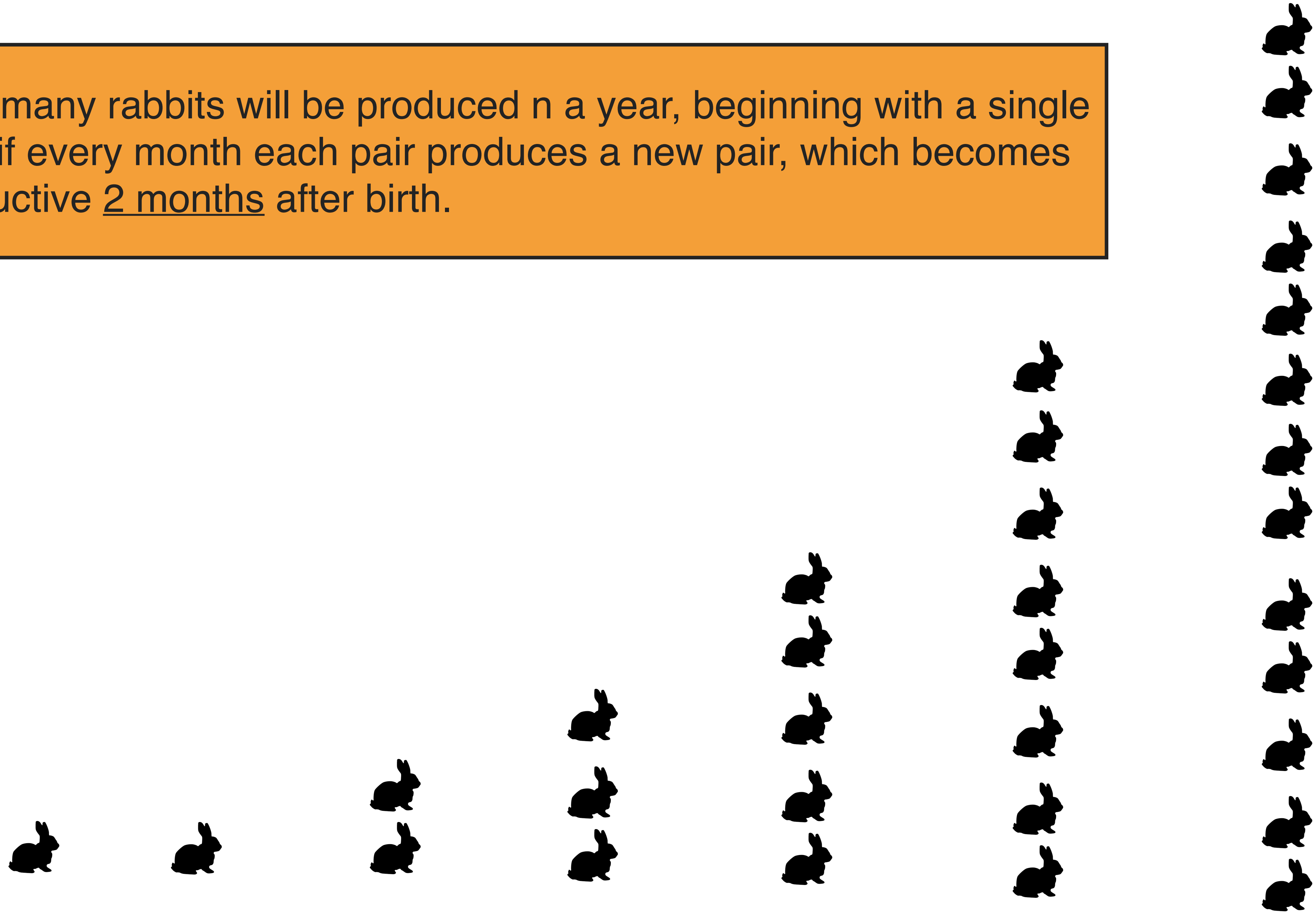
$$\boxed{A^n = S(\Lambda^n)S^{-1}}$$

Application

How many rabbits will be produced n a year, beginning with a single pair, if every month each pair produces a new pair, which becomes productive <u>2 months</u> after birth.

due to Leonardo Pissa, circa 12th century

How many rabbits will be produced n a year, beginning with a single pair, if every month each pair produces a new pair, which becomes productive 2 months after birth.



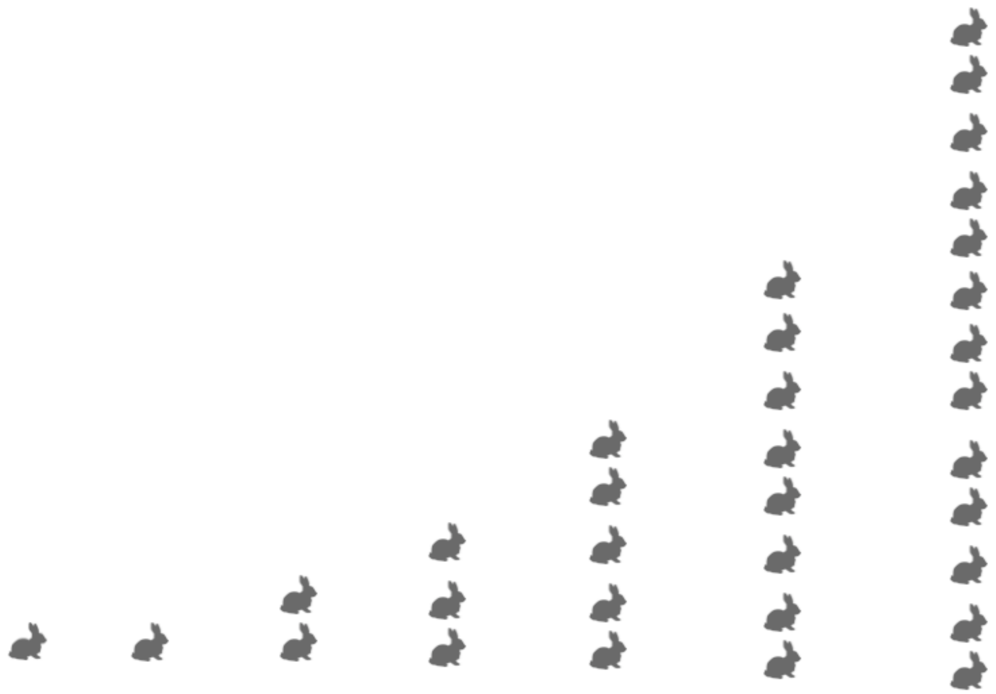
$x_1 = 1$ $x_2 = 1$ $x_3 = 2$ $x_4 = 3$ $x_5 = 5$ $x_6 = 8$ $x_7 = 13$ $x_8 = 21$

$$x_n = x_{n-1} + x_{n-2}$$

First order difference equations

A **difference equation** is a rule that expresses a sequence, in terms of previous members of the sequence (starting from some initial values)

A difference equation is **first order** if the k^{th} member is defined in terms of the the $(k-1)^{th}$ member.



$$y_k = y_{k-1} + y_{k-2}$$

$$y_0 = y_1 = 1$$

Second order

Is there a way to <u>transform</u> this system into a <u>first order</u> system?

$$x_{k+1} = x_k + x_{k-1}$$

$$\underbrace{x_0 \quad x_1}_{U_0} \quad \underbrace{x_2 \quad x_3}_{U_2} \quad \cdots\cdots \quad \underbrace{x_{k-1} \quad x_k}_{U_{k-1}} \quad x_{k+1} \quad \cdots\cdots$$

$$x_0 \quad \underbrace{x_1 \quad x_2}_{U_1} \quad x_3 \quad \cdots\cdots \quad x_{k-1} \quad \underbrace{x_k \quad x_{k+1}}_{U_k} \quad \cdots\cdots$$

?

$$U_k = A U_{k-1}$$

$$\begin{pmatrix} x_{k+1} \\ x_k \end{pmatrix} = A \begin{pmatrix} x_k \\ x_{k-1} \end{pmatrix}$$

$$x_{k+1} = a_{00} x_k + a_{01} x_{k-1}$$
$$x_k = a_{10} x_k + a_{11} x_{k-1}$$

$$\begin{pmatrix} x_{k+1} \\ x_k \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} \begin{pmatrix} x_k \\ x_{k-1} \end{pmatrix}$$

$$\begin{pmatrix} x_{k+1} \\ x_k \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x_k \\ x_{k-1} \end{pmatrix}$$

$$U_k = AU_{k-1}$$

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

Finding the n^{th} term in the sequence….

$$U_1 = AU_0$$

$$U_2 = AU_1 = A^2U_0$$

$$U_3 = AU_2 = A^3U_0$$

$$\vdots$$

$$U_n = AU_{n-1} = A^nU_0$$

Using the eigendecomposition theorem:

$$U_n = S\Lambda^n S^{-1}U_0$$

numpy.linalg.eig

`numpy.linalg.eig`(*a*) [source]

Compute the eigenvalues and right eigenvectors of a square array.

Parameters: **a** : *(..., M, M) array*

Matrices for which the eigenvalues and right eigenvectors will be computed

Returns: **w** : *(..., M) array*

The eigenvalues, each repeated according to its multiplicity. The eigenvalues are not necessarily ordered. The resulting array will be of complex type, unless the imaginary part is zero in which case it will be cast to a real type. When *a* is real the resulting eigenvalues will be real (0 imaginary part) or occur in conjugate pairs

v : *(..., M, M) array*

The normalized (unit “length”) eigenvectors, such that the column `v[:,i]` is the eigenvector corresponding to the eigenvalue `w[i]`.

Raises: **LinAlgError**

If the eigenvalue computation does not converge.

See also:

`eigvals` eigenvalues of a non-symmetric array.

`eigh` eigenvalues and eigenvectors of a symmetric or Hermitian (conjugate symmetric) array.

`eigvalsh` eigenvalues of a symmetric or Hermitian (conjugate symmetric) array.

https://docs.scipy.org/doc/numpy-1.14.0/reference/generated/numpy.linalg.eig.html

Discussion

- Today’s discussion was centred on modelling, first order system.
- In practice, most methods to compute eigenvectors are iterative.
- Eigensystems will appear again later in the unit.