EOSC 213: Computational Methods in Geological Engineering

Lecture 10: Systems of ODEs + matrix forms + phase space diagrams

Shadab Ahamed

Department of Earth, Ocean and Atmospheric Sciences,
The University of British Columbia

Thu, Feb 5, 2026

Course Logistics

Homework 02

- Homework 02 has been posted on Canvas
- Submission deadline: **Friday, 20th Feb, 2026** at **11:59 pm**
- This homework will be good practice for the upcoming midterm on 24th Feb, 2026
- Please follow the submission instructions carefully:
  - Submit the **written solutions as a single PDF**
  - For the Python notebook, submit:
    - the `.ipynb` file
    - the corresponding `.html` export (TA wants to award -2 if this is missing)

Office Hour

- Today's Office Hour: 2:00 pm - 3:00 pm, **first floor of the ESB**

Recap: Ordinary differential equations (ODEs)

A first order ODE describes how a quantity changes in time,

$$\frac{dx(t)}{dt} = f(t, x(t))$$

- $x(t)$: unknown function (state)
- $f(t, x(t))$: rule that gives the rate of change
- initial condition: $x(t_0) = x_0$

Goal: predict $x(t)$ for $t > t_0$

Recap: Time discretization

We approximate solutions at discrete times,

$$t_j = t_0 + jh, \qquad j = 0, 1, 2, \ldots$$

Store,

$$x_j \approx x(t_j)$$

Numerical methods are **one-step updates**,

$$x_{j+1} = \Phi(h, t_j, x_j)$$

Key idea: turn a differential equation into a recurrence relation.

Recap: Forward Euler method

Given:

$$\frac{dx}{dt} = f(t, x)$$

Forward Euler update:

$$x_{j+1} = x_j + h\, f(t_j, x_j)$$

- uses slope at the **current** point
- first-order accurate
- conditionally stable

Important: too large h can cause oscillation or blow-up.

Recap: Explicit Midpoint method

Two-step idea:

Half-step predictor:

$$x_{j+\frac{1}{2}} = x_j + \frac{h}{2} f(t_j, x_j)$$

Full update:

$$x_{j+1} = x_j + h\, f\left(t_j + \frac{h}{2}, x_{j+\frac{1}{2}}\right)$$

- second-order accurate
- better behavior than Forward Euler
- still **conditionally stable**

Recap: Backward Euler method

Backward Euler update:

$$x_{j+1} = x_j + h\,f(t_{j+1}, x_{j+1})$$

- implicit: x_{j+1} appears on both sides
- requires solving an equation each step
- **unconditionally stable** for decay problems

Key lesson: stability $\neq$ accuracy.

Lecture today

Today we begin a new topic:

- **Systems of ordinary differential equations**
- Writing systems using vectors and matrices
- Some basic math about eigenvalues of matrices

This lecture builds the foundation for:

- stability analysis
- stiffness
- numerical behavior of time-stepping methods for system of ODEs

Learning goals

By the end of this lecture, you should be able to:

- write a system of two coupled ODEs
- rewrite the system using vector and matrix notation
- compute eigenvalues of a 2×2 matrix

Review: a single ODE

A single ODE looks like,

$$\frac{dx}{dt} = f(x, t)$$

This describes:

- one unknown function $x(t)$
- evolving in time

What changes when we have two variables changing in time?

Suppose we now track two quantities,

$$x(t), \quad y(t)$$

Each quantity can potentially influence the other.

A system of two ODEs

A system has the form,

$$\frac{dx}{dt} = f(x, y, t),$$
$$\frac{dy}{dt} = g(x, y, t)$$

- two equations
- two functions f and g
- equations could be **coupled**

What does “coupled” mean?

Coupling means:

- $x'(t)$ may depend on $y(t)$
- $y'(t)$ may depend on $x(t)$

This interaction is what makes systems *interesting and harder to solve*.

Why we need systems of ODEs

Many real problems involve **multiple interacting quantities**:

- position and velocity
- temperature and energy
- multiple chemical species
- coupled mechanical components

Tracking just one variable is often not enough.

Example 1: Two independent decays

Consider the model,

$$x' = -x,$$
$$y' = -2y$$

- x and y decay independently
- no coupling between variables
- Exact analytical solutions: $x(t) = x_0 e^{-t}$ and $y(t) = y_0 e^{-2t}$

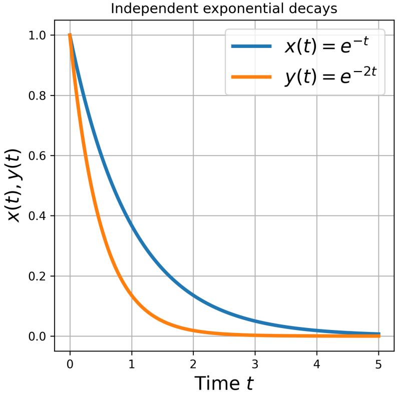


Figure: $x(t)$ and $y(t)$ decay exponentially, without influencing each other.

Example 1: Two independent decays (cont'd)

Consider the model,

$$x' = -x,$$
$$y' = -2y$$

- $x(t) \to 0$ as $t \to \infty$
- $y(t) \to 0$ faster than $x(t)$ as $t \to \infty$
- Steady state: $x' = y' = 0 \implies x = 0, y = 0$

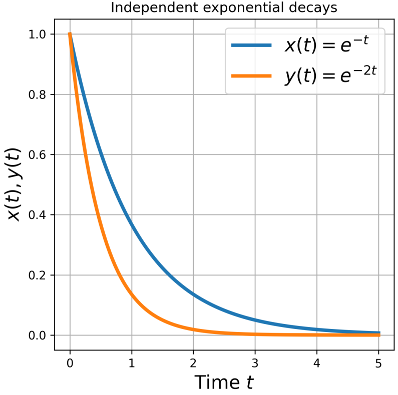


Figure: $x(t)$ and $y(t)$ decay exponentially, without influencing each other.

Example 2: Weak coupling

Consider the model,

$$x' = -x + y,$$
$$y' = -2y$$

- y influences the evolution of x
- y by itself is independent
- y still decays
- While y is non-zero, it *drives* x

Now y influences the evolution of x.

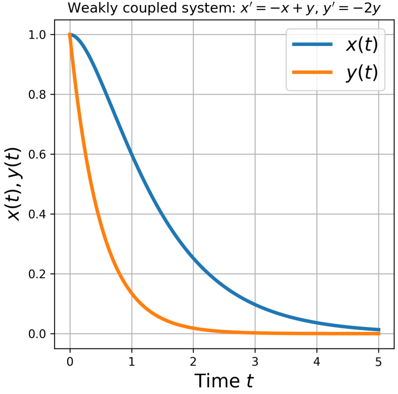


Figure: Both $x(t)$ and $y(t)$ decay, but $y(t)$ influences $x(t)$ whenever $y(t) \neq 0$.

Example 3: Oscillatory coupling

Consider the model,

$$
\begin{aligned}
x' &= y, \\
y' &= -x
\end{aligned}
$$

- Neither x nor y decays directly; there is no damping term in the system.
- Each variable acts as the *rate of change* of the other, creating a continuous feedback loop.
- When x is positive, it drives y downward; when y is positive, it drives x upward.

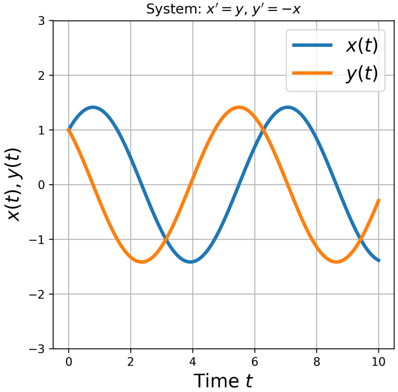


Figure: Time evolution of $x(t)$ and $y(t)$. Neither variable decays directly; instead, they continuously drive each other, producing oscillatory motion with constant amplitude.

Example 4: A system of three coupled ODEs

Consider the model,

$$
\begin{aligned}
x' &= -x + y, \\
y' &= -y + z, \\
z' &= -z
\end{aligned}
$$

- The system now has **three state variables**: (x, y, z)
- z decays independently
- y is driven by z, and x is driven by y
- Information flows sequentially: $z \to y \to x$

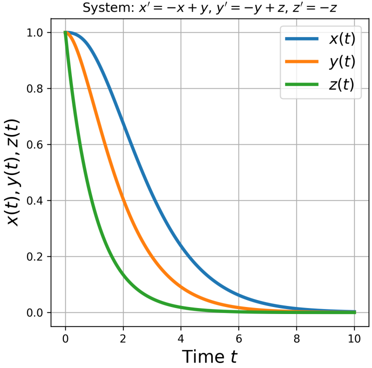


Figure: Time evolution of $x(t)$, $y(t)$, and $z(t)$. The variable z decays independently, while y and x are successively driven by downstream variables, producing delayed but monotone decay toward the equilibrium at $(0, 0, 0)$.

Example 5: A system of six coupled ODEs

Consider a model with six interacting variables

$$x_1(t),\ x_2(t),\ x_3(t),\ x_4(t),\ x_5(t),\ x_6(t)$$

One possible model is,

$$\begin{aligned}
x_1' &= -2x_1 + 0.5x_2 + x_3 \\
x_2' &= 0.3x_1 - x_2 + 0.7x_4 \\
x_3' &= -0.2x_1 + 0.5x_2 - x_3 + x_5 \\
x_4' &= 0.4x_2 - 1.5x_4 + 0.6x_5 \\
x_5' &= 0.2x_3 + 0.8x_4 - x_5 + x_6 \\
x_6' &= 0.5x_4 - 2x_6
\end{aligned}$$

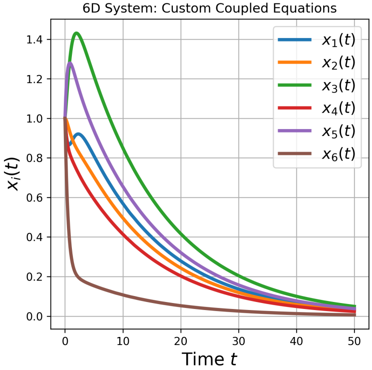


Figure: Time evolution of six coupled variables $x_1(t), \ldots, x_6(t)$ for a linear system with multiple cross-dependencies and distinct coefficients. Although all variables decay, the coupling redistributes influence across components, producing different transient behaviors and time scales.

Handling many equations quickly becomes cumbersome

Writing many equations becomes cumbersome. We need better representations to work with and compute these systems.

“Vector notation” comes to the rescue!

Vector notation:

- is compact
- matches how we code things
- reveals mathematical structure

State vector

Let us define,

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

This single (column) vector represents the full state.

Vector form of a system

The system can be represented by,

$$\mathbf{x}'(t) := \frac{d}{dt}\mathbf{x}(t) = \begin{bmatrix} x_1'(t) \\ x_2'(t) \end{bmatrix} = \mathbf{F}(\mathbf{x}, t)$$

where,

$$\mathbf{F}(\mathbf{x}, t) = \begin{bmatrix} f(x_1, x_2, t) \\ g(x_1, x_2, t) \end{bmatrix}$$

Vector form of a system

The system can be represented by,

$$\mathbf{x}'(t) := \frac{d}{dt}\mathbf{x}(t) = \begin{bmatrix} x_1'(t) \\ x_2'(t) \end{bmatrix} = \mathbf{F}(\mathbf{x}, t)$$

where,

$$\mathbf{F}(\mathbf{x}, t) = \begin{bmatrix} f(x_1, x_2, t) \\ g(x_1, x_2, t) \end{bmatrix}$$

- numerical solvers operate on vectors
- theory is written using vectors
- higher dimensions work the same way

A 1000-variable system looks identical in notation.

Linear systems: from equations to matrices

Many important systems can be written as a **linear** system,

$$\mathbf{x}'(t) = A\,\mathbf{x}(t)$$

where A is a constant matrix.

In this form:

- $\mathbf{x}(t)$ is the **state vector**
- A encodes all **coupling coefficients**
- the system is compact and scalable to large dimension

2D case: what does $\dot{\mathbf{x}} = A\mathbf{x}$ mean?

Let

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}, \qquad A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Then,

$$\mathbf{x}' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax + by \\ cx + dy \end{bmatrix}$$

So, $\dot{\mathbf{x}} = A\mathbf{x}$ is just a compact way to write,

$$x' = ax + by, \qquad y' = cx + dy.$$

Phase portraits: Plotting $y(t)$ vs. $x(t)$

For a system of ODEs with two variables,

$$x' = f(x, y),$$
$$y' = g(x, y),$$

we can visualize solutions in the **phase plane**.

A **phase portrait** is a plot of:

- $y(t)$ vs. $x(t)$
- not variables versus time

Each point (x, y) represents the **state of the system** at some time.

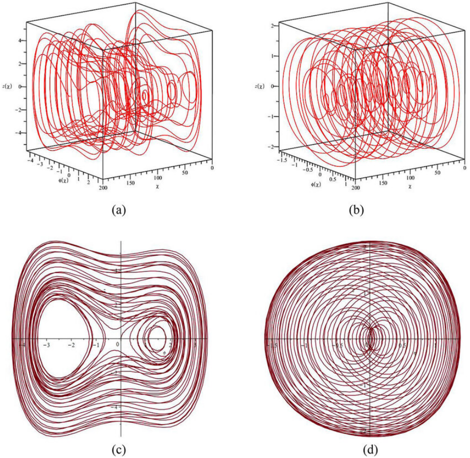


Figure: C. Liu, *The chaotic behavior and traveling wave solutions of the conformable extended Korteweg-de-Vries model*, **Open Physics**, Volume 22 Issue 1

How to read a phase portrait?

In a phase portrait:

- each curve is a solution trajectory
- each trajectory corresponds to a different initial condition
- arrows (or direction) indicate how the state evolves in time

Time is **implicit**:

- we do not see t on the axes
- we see the *path* taken by the system in state space

Phase portraits show **qualitative behavior**, not exact values.

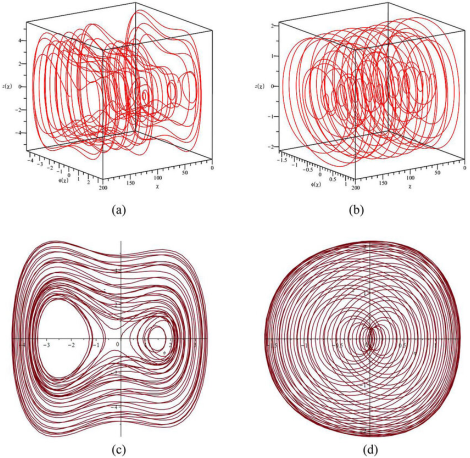


Figure: C. Liu, *The chaotic behavior and traveling wave solutions of the conformable extended Korteweg-de-Vries model*, **Open Physics**, Volume 22 Issue 1

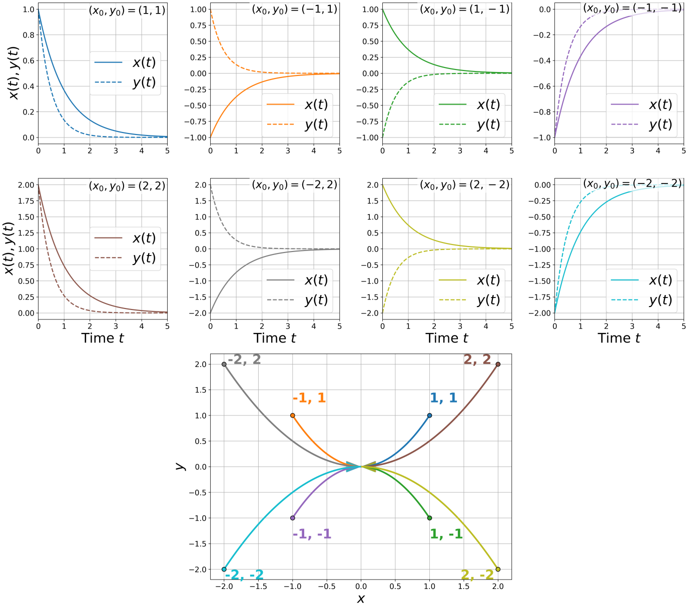
Example of phase portrait for different initial conditions

Consider the two independent decays model from Example 1,

$$x' = -x,$$
$$y' = -2y$$

for 8 different initial conditions:

$$(x_0, y_0) = \{(1,1), (-1,1), (1,-1), (-1,-1)\\ (2,2), (-2,2), (2,-2), (-2,-2)\}$$



Example 1 in matrix form: Two independent decays

Recall,

$$x' = -x, \qquad y' = -2y$$

Define the state vector,

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

Then,

$$\mathbf{x}' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix}}_{A} \begin{bmatrix} x \\ y \end{bmatrix}$$

- diagonal entries control self decay
- off-diagonal entries are zero $\Rightarrow$ no coupling

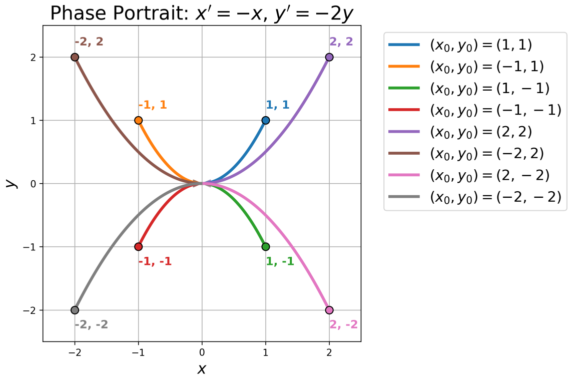


Figure: Phase portrait of the uncoupled decay system in Example 1. Trajectories from different initial conditions decay monotonically toward the stable equilibrium at the origin (0, 0).

Example 2 in matrix form: Weak coupling

Recall,

$$x' = -x + y, \qquad y' = -2y$$

Let

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

Then,

$$\mathbf{x}' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} -1 & 1 \\ 0 & -2 \end{bmatrix}}_{A} \begin{bmatrix} x \\ y \end{bmatrix}$$

- $A_{12} = 1$ means y drives x
- $A_{21} = 0$ means x does not drive y

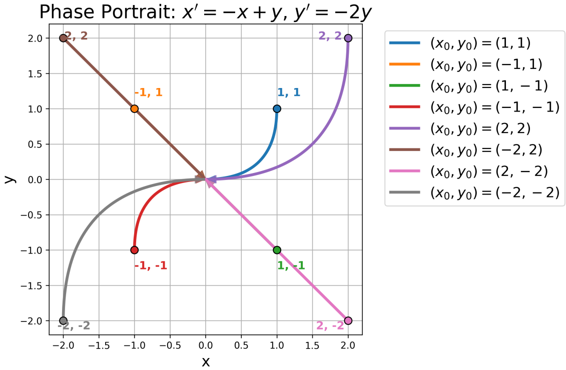


Figure: Phase portrait of the weakly coupled system in Example 2. All trajectories converge to the stable equilibrium at the origin (0,0).

Example 3 in matrix form: Oscillatory coupling

Recall,

$$x' = y, \qquad y' = -x$$

Let

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

Then,

$$\mathbf{x}' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}}_{A} \begin{bmatrix} x \\ y \end{bmatrix}$$

- purely off-diagonal coupling
- no negative diagonal "decay" terms $\Rightarrow$ no damping

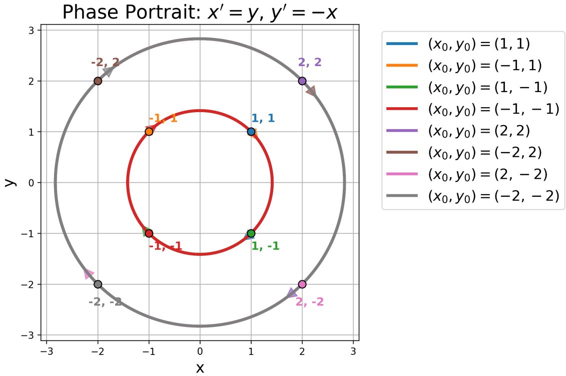


Figure: Phase portrait of the oscillatory system in Example 3. Trajectories form closed orbits around the origin $\Longrightarrow$ sustained oscillations with fixed amplitude. The equilibrium at $(0, 0)$ is neutrally stable.

Example 4 in matrix form: A system of three coupled ODEs

Recall, $x' = -x + y, \quad y' = -y + z, \quad z' = -z$

Let

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Then,

$$\mathbf{x}' = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \underbrace{\begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{bmatrix}}_{A} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

- still linear: coefficients are constants
- upper-triangular structure reflects the chain $z \to y \to x$

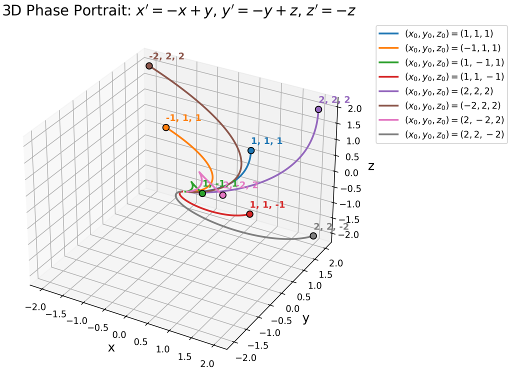


Figure: 3D phase portrait of the coupled system in Example 4. All trajectories converge to the stable equilibrium at the origin, but coupling causes the motion to bend through three-dimensional state space.

Example 5 in matrix form: A system of six coupled ODEs

Recall the six coupled equations,

$$
\begin{aligned}
x_1' &= -2x_1 + 0.5x_2 + x_3 \\
x_2' &= 0.3x_1 - x_2 + 0.7x_4 \\
x_3' &= -0.2x_1 + 0.5x_2 - x_3 + x_5 \\
x_4' &= 0.4x_2 - 1.5x_4 + 0.6x_5 \\
x_5' &= 0.2x_3 + 0.8x_4 - x_5 + x_6 \\
x_6' &= 0.5x_4 - 2x_6
\end{aligned}
$$

Define,

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

Then the system can be written compactly as,

$$\mathbf{x}'(t) = A\,\mathbf{x}(t)$$

where A is the 6×6 matrix of coefficients (shown on the next slide).

Example 5 in matrix form: A system of six coupled ODEs (cont'd)

For the same system, the 6×6 matrix A is,

$$A = \begin{bmatrix} -2 & 0.5 & 1 & 0 & 0 & 0 \\ 0.3 & -1 & 0 & 0.7 & 0 & 0 \\ -0.2 & 0.5 & -1 & 0 & 1 & 0 \\ 0 & 0.4 & 0 & -1.5 & 0.6 & 0 \\ 0 & 0 & 0.2 & 0.8 & -1 & 1 \\ 0 & 0 & 0 & 0.5 & 0 & -2 \end{bmatrix}$$

- each row corresponds to one equation (x_i')
- each entry is the coefficient multiplying x_j
- writing $\mathbf{x}' = A\mathbf{x}$ automatically represents all 6 equations at once

Why matrix form is a big deal

- Instead of writing n equations, we write only one matrix equation,

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

- The structure of the system is visible in A:
  - diagonal terms: self-decay / self-growth
  - off-diagonal terms: coupling
  - matrix sparsity patterns: who talks to whom

- This is the starting point for:
  - eigenvalues and modes
  - stability of continuous-time systems
  - numerical stability of time-stepping methods on systems

Digression into some Math: Eigenvalues and Eigenvectors

An *eigenvector* $\mathbf{v}$ of a matrix A satisfies,

$$A\mathbf{v} = \lambda\mathbf{v},$$

where λ is the *eigenvalue*.

We shall look into the **geometric meaning** of eigenvectors and eigenvalues in the next lecture. For now, let's learn how to compute the eigenvalues.

Computing eigenvalues

Eigenvalues satisfy,

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

This produces a polynomial equation called the *characteristic equation*.

Eigenvalues of a 2×2 matrix: computing the characteristic equation

Consider the matrix,

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Eigenvalues λ are defined by the condition,

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

First compute $A - \lambda I$,

$$A - \lambda I = \begin{pmatrix} a - \lambda & b \\ c & d - \lambda \end{pmatrix}$$

Now take the determinant,

$$\det(A - \lambda I) = (a - \lambda)(d - \lambda) - bc$$

Eigenvalues of a 2×2 matrix: trace and determinant

Expanding,

$$\lambda^2 - (a+d)\lambda + (ad - bc) = 0$$

This is a quadratic equation in λ whose solution determines the eigenvalues.

For the matrix $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, let us define,

$$\text{tr}(A) = \text{ sum of diagonal elements } = a + d,$$

And,

$$\det(A) = ad - bc$$

Eigenvalues of a 2×2 matrix: trace and determinant

$$\mathrm{tr}(A) = a + d,$$

$$\det(A) = ad - bc$$

The solutions of the quadratic equation are,

$$\lambda^2 - (a+d)\lambda + (ad - bc) = 0$$

$$\lambda_{1,2} = \frac{(a+d) \pm \sqrt{(a+d)^2 - 4(ad - bc)}}{2}$$

$$\lambda_{1,2} = \frac{\mathrm{tr}(A) \pm \sqrt{\mathrm{tr}(A)^2 - 4\det(A)}}{2}$$

Computing eigenvectors

For each eigenvalue λ,

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

Solve this linear system to get $\mathbf{v}$.

Example: Computing eigenvalues and eigenvectors

Consider the weakly coupled system from Example 2,

$$\begin{aligned} x' &= -x + y, \\ y' &= -2y \end{aligned}$$

The corresponding matrix form for this system is,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

So,

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

Let us compute its eigenvalues and eigenvectors step-by-step.

Example: Computing eigenvalues and eigenvectors

Compute the characteristic equation,

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

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

Taking the determinant,

$$\det(A - \lambda I) = (-1-\lambda)(-2-\lambda)$$

So the characteristic equation is,

$$(-1-\lambda)(-2-\lambda) = 0$$

Example: Computing eigenvalues and eigenvectors

Solving,

$$(-1-\lambda)(-2-\lambda) = (\lambda+1)(\lambda+2) = 0$$

We get the eigenvalues,

$$\lambda_1 = -1, \qquad \lambda_2 = -2$$

- Both eigenvalues are real and negative
- Different magnitudes imply different decay rates

Example: Computing eigenvalues and eigenvectors

Computing the eigenvector for the eigenvalue $\lambda_1 = -1$, let us solve,

$$(A - \lambda_1 I) = 0 \implies (A - (-1)I) = (A + I)\mathbf{v}_1 = \left(\begin{bmatrix} -1 & 1 \\ 0 & -2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

This gives the system,

$$y_1 = 0$$

So one eigenvector is,

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Example: Computing eigenvalues and eigenvectors

Computing the eigenvector for the eigenvalue $\lambda_2 = -2$, let us solve,

$$(A - \lambda_2 I) = 0 \implies (A - (-2)I) = (A + 2I)\mathbf{v}_2 = \left(\begin{bmatrix} -1 & 1 \\ 0 & -2 \end{bmatrix} + 2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

This gives the system,

$$x_2 = -y_2$$

So one eigenvector is,

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

Summary: What we learned today

In this lecture, we

- moved from single ODEs to **systems of ODEs**
- saw how coupling between variables changes system behavior
- used **phase portraits** to visualize dynamics qualitatively
- learned how vector and matrix notation compactly represent systems
- introduced **eigenvalues and eigenvectors** as key mathematical tools (we will need these in the next lecture to understand the dynamics of the system)

Big picture:

$$\text{systems} \;\rightarrow\; \text{vectors} \;\rightarrow\; \text{matrices} \;\rightarrow\; \text{eigenvalues}$$

These ideas form the foundation for understanding stability and numerical behavior.

Why we needed all this

Next lectures:

- numerical methods create discrete systems
- matrices become amplification matrices
- eigenvalues determine numerical stability

What we will do next

In the next lecture, we will

- study the **geometric meaning** of eigenvalues and eigenvectors
- connect eigenvalues to:
  - decay and growth
  - oscillations
  - stability of equilibria
- use eigenvectors to understand the **shape of phase portraits**
- prepare for stability analysis of **numerical methods on systems**

Goal: predict system behavior without explicitly solving the ODEs.