

CONTINUOUS-TIME DIFFERENTIAL EQUATIONS MODELS (II) (DYNAMICAL SYSTEMS)

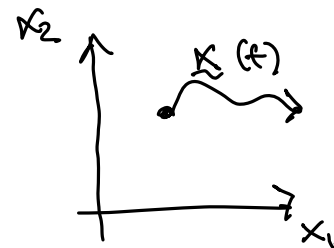
Today: Ch. 5.1-5.2, Sec 1.4.1, of Etc.

- State vector. $x(t)$.
- Evolves in phase space

E.g. $x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$; $x_1(t)$ = concentration of protein
 $x_2(t)$ = concentration of RNA

phase space = plane

Two-dimensional



- Simplest example: one-dimensional phase space
 $\tilde{x}(t) = x(t)$

"•" = " $\frac{d}{dt}$ " $\rightarrow \dot{x} = \frac{dx}{dt} = f(x)$

Notation

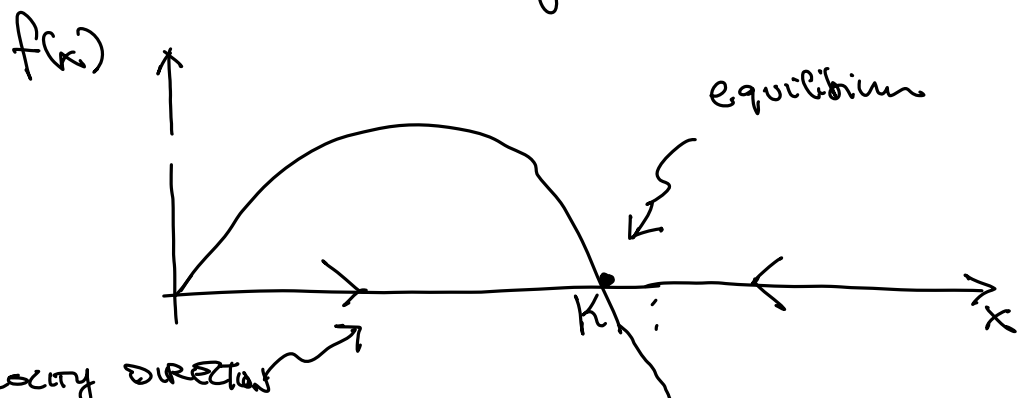
Ex. $x(t)$ = number of cells

logistic model of cell growth:

$$\dot{x} = x \cdot r \left(1 - x/k \right)$$

per-capita growth rate

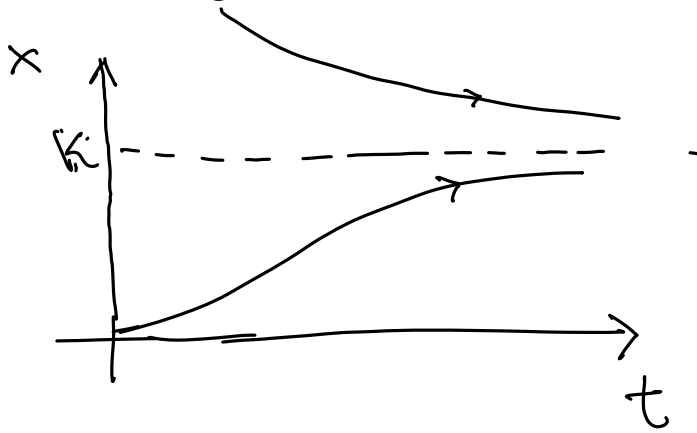
per-capita growth rate



SKETCH VELOCITY DIRECTION

• Def: \bar{x} is equilibrium if $f(\bar{x}) = 0$.

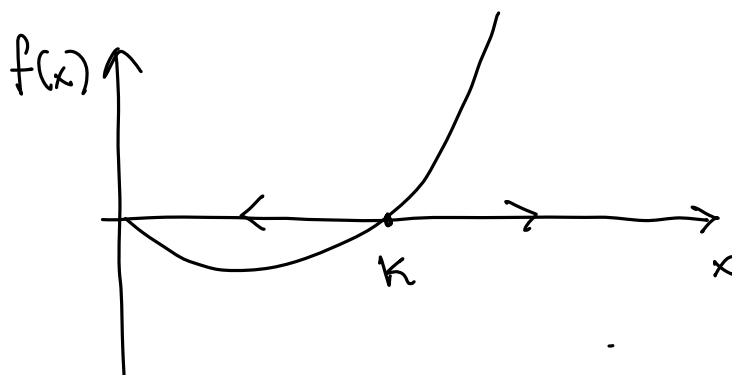
• From Geometry immediately sketch qualitative dynamics:



• definition: a solution $x(t)$ to $\dot{x} = f(x)$ with $x(0) = x_0$
 is a trajectory from x_0
 (Two trajectories shown above.)

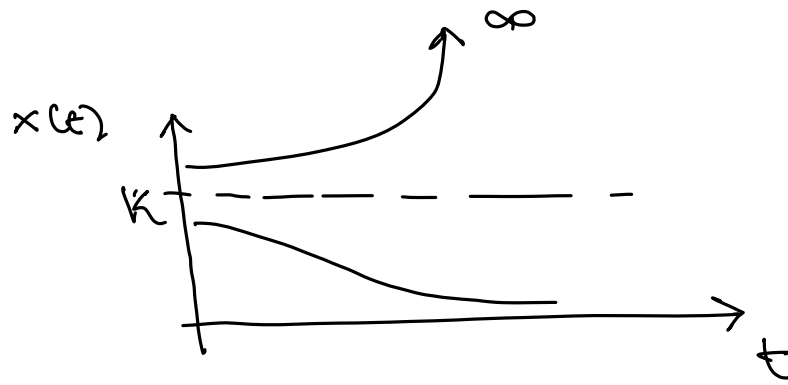
Ex 2 | "Superlinear population growth" - Eng. increased growth rate w/ probab. finding water (s).

$$\dot{x} = r x \left(-1 + \frac{x}{k} \right)$$

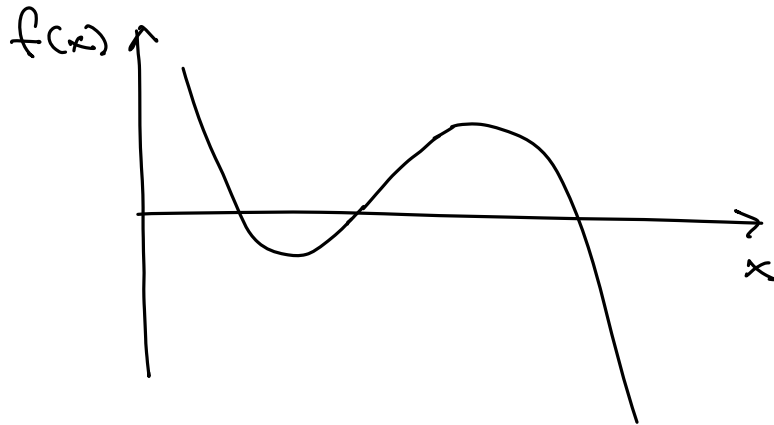


Fact: $x(t) \rightarrow \infty$
 in FINITE TIME

Discuss re:
 Edelstein-Keshet
 M. Crichton
Quote!



Ex. 3



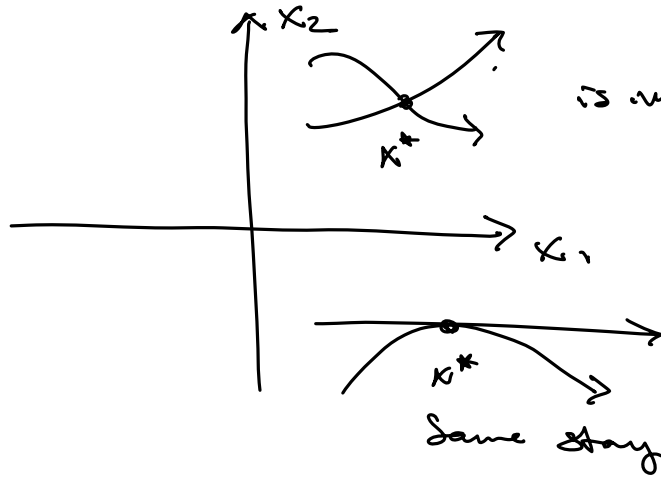
[Sketch trajectories $x(t)$ on board ...]

- STATE VECTORS (AND PHASE SPACES) OF TWO- AND HIGHER-DIMENSION :

$$\frac{d\tilde{x}}{dt} = \begin{pmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \vdots \\ \frac{dx_n}{dt} \end{pmatrix} = \tilde{f}(\tilde{x}) = \begin{pmatrix} f_1(x_1, x_2, \dots, x_n) \\ f_2(x_1, x_2, \dots, x_n) \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) \end{pmatrix}$$

Theorem: Let $f(x)$ be continuously differentiable (for $x \in U$)
 Then, for any $x_0 \in U$, trajectories from x_0
 exist and are unique for some time period
 $t \in [0, \tau)$.

Result:



For strange examples when $f(x)$ is Not differentiable, try

$$\dot{x} = \sqrt{x} \quad \text{from } x_0 = 0$$

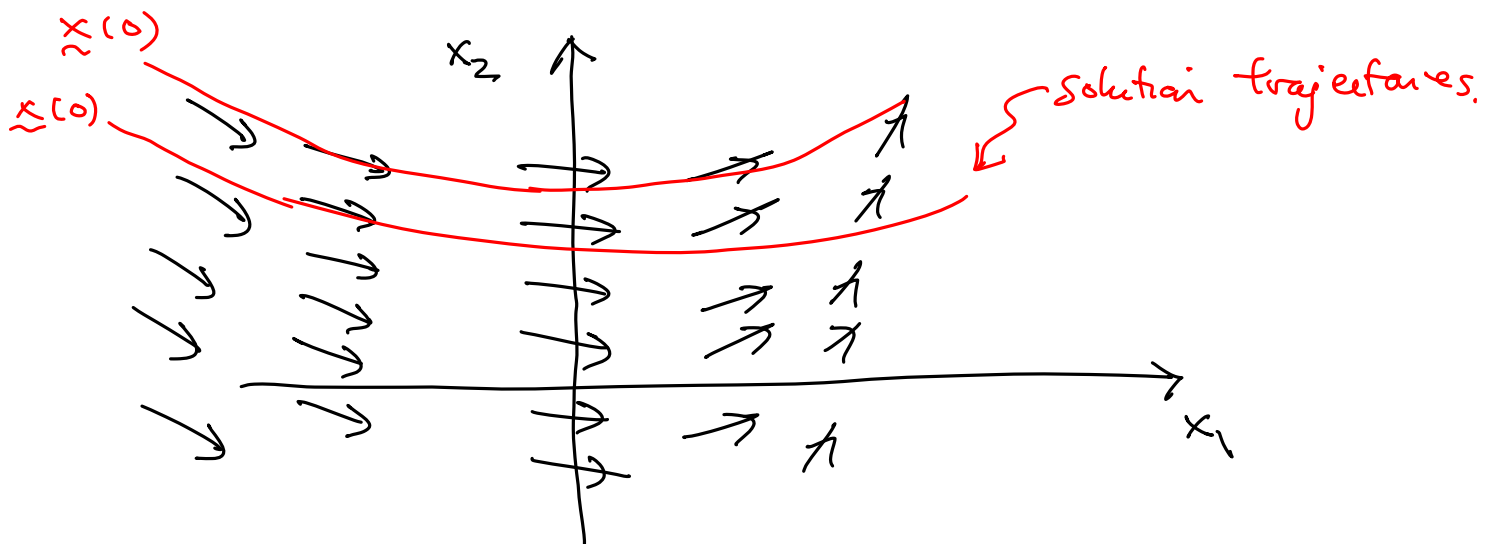
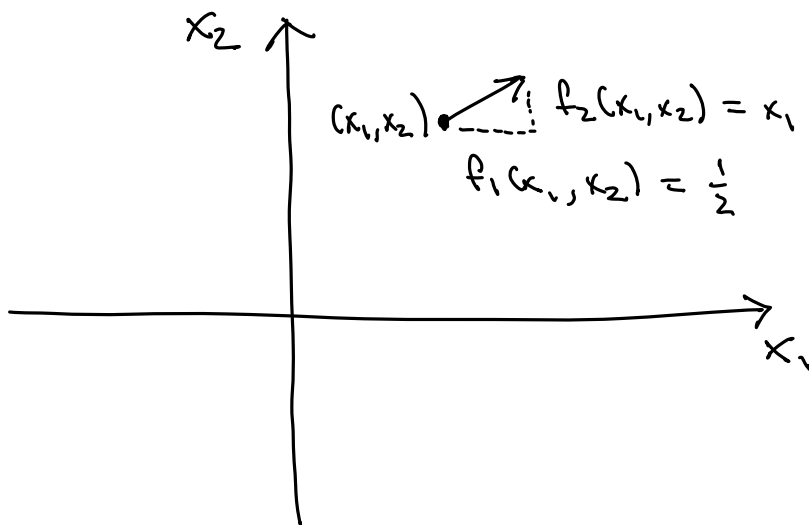
$$x_\tau(t) = \begin{cases} 0 \\ \frac{1}{4}(t-\tau)^2 \end{cases} \quad \text{is solution for any } \tau \geq 0.$$

Visualize for $n=2$ dimensions.

phase space = phase plane

E.g.
$$\begin{cases} dx_1/dt = 1/2 = f_1(x_1, x_2) \\ dx_2/dt = x_1 = f_2(x_1, x_2) \end{cases}$$

Plot "direction field" [quiver plot]



• Dynamical systems studies trajectories that arise from following direction fields $f(\underline{x})$.

Euler's Method...

$$f_1(x_1, x_2)$$

$$\frac{dx_2}{dt} = f_2(x_1, x_2)$$

trajectory "at" $(x_1(t), x_2(t))$

Where is it at $t + \Delta t$?

RATE OF CHANGE OF $x_1 = f_1(x_1(t), x_2(t))$
ELAPSED TIME = Δt

$$\rightarrow x_1(t + \Delta t) = x_1(t) + \underbrace{f_1(x_1(t), x_2(t))}_{\text{RATE}} \times \underbrace{\Delta t}_{\text{TIME}}$$

DISTANCE =

$$x_2(t + \Delta t) = \dots$$

That's what we're saying in direction field plot —

(RETURN TO DIRⁿ FIELD Plot, SCALE By Δt)

(More on numerical timestepping schemes next time.)

- When dimension $n \geq 2$, ~~Extremely~~ complex and beautiful trajectories (dynamics) occur... for simple-looking differential equations

Ex 11

$$\frac{dx_1}{dt} = \sigma(x_2 - x_1)$$

$$\frac{dx_2}{dt} = x_1(\rho - x_3) - x_2$$

$$\frac{dx_3}{dt} = x_1 x_2 - \beta x_3$$

Lorenz system.

MATLAB DEMO

lorenz.m

NOTE:

Can also define: TIME-DEPENDENT velocity fields.

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

In class, just list 3 steps on board— cover in lab assignment!

Analyzing differential equations in MATLAB.

- MAKING DIRECTION FIELD PLOTS.

1) Define function file for $\vec{f}(\vec{x})$

MATLAB: an "odefun"

describe system
there!

• Suggested filename: \swarrow lorenz-odefun.m

• Syntax.

always include time

function $dxdt = \text{lorenz_odefun}(t, x)$

↓
↑
vector

$$x1 = x(1)$$

$$x2 = x(2)$$

$$x3 = x(3)$$

$$\text{sigma} = 10$$

$$\text{rho} = 28$$

$$\text{beta} = 8/3$$

$$dx1dt = \text{sigma} * (x1 - x2)$$

$$dx2dt = x1 * (\text{rho} - x3) - x2$$

$$dx3dt = x1 * x2 - \text{beta} * x3$$

$$dxdt = [dx1dt ;$$

$$dx2dt ;$$

$$dx3dt]$$

← RETURNS A
COLUMN VECTOR.

OR

function $dxdt = \text{simple_odefun}(t, x)$

$$x1 = x(1)$$

$$x2 = x(2)$$

$$dx1dt = 1/2$$

$$dx2dt = x1$$

$$dxdt = [dx1dt ; dx2dt]$$

[fill in:
class]

(Assume $n=2$ dimensions)

2) Make "mesh" of x_1, x_2 values and plot "direction arrows"
"quiver plot"

See:

CODE

direction-field-plotter.m

~~~~~

Please note COMMENTS in this code, explaining :

- meshgrid
- quiver commands

3) Apply Euler's method  $\rightarrow$  sol<sup>n</sup> trajectory.