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

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

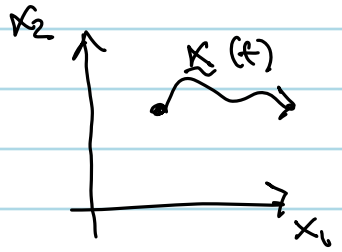
- State vector. $x(t)$.
- Evolves in phase space

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

$\underset{\sim}{x}(t) = x(t)$

2011 TEACHING: SKIP FROM HERE ALL THE WAY TO DIRECTION FIELD

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

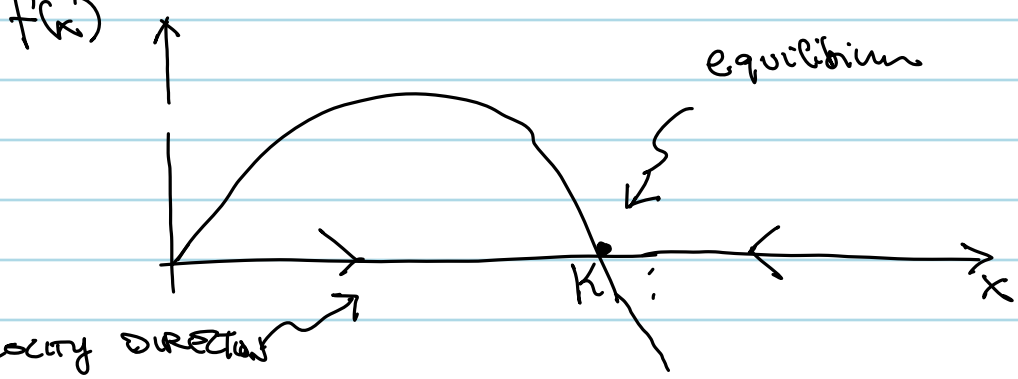
NOTATION

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

logistic model of cell growth:

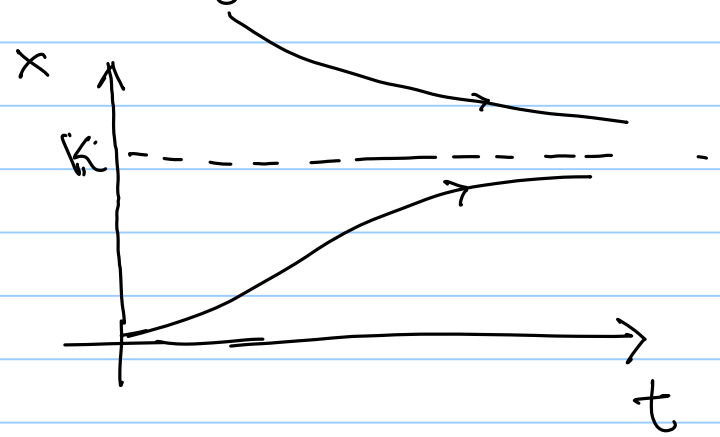
$$\dot{x} = x \cdot r\left(1 - x/k\right)$$

per-capita growth rate



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

- From geometry immediately sketch qualitative dynamics:

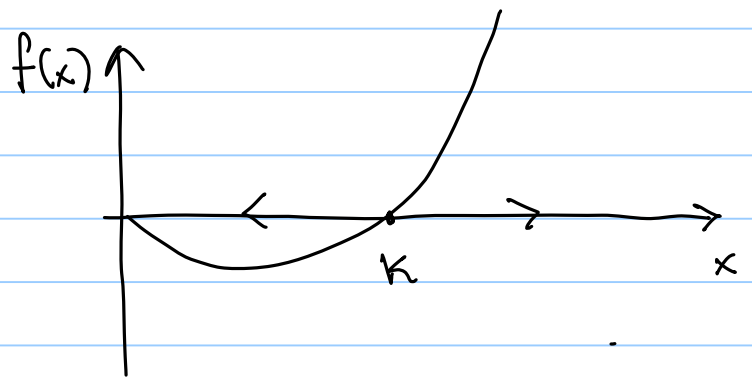


- definition: a solution $x(t)$ to $\dot{x} = f(x)$ with $x(0) = x_0$ is a trajectory from x_0

(Two trajectories shown above.)

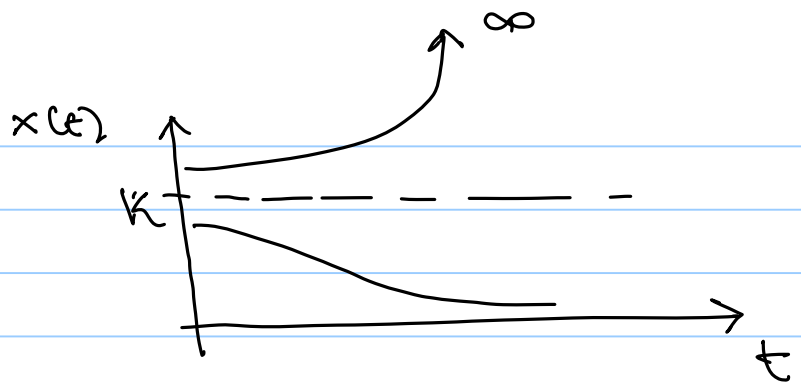
Ex 2] "Superlinear population growth" – E.g. increased growth rate w/ prob. funding rate (?).

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

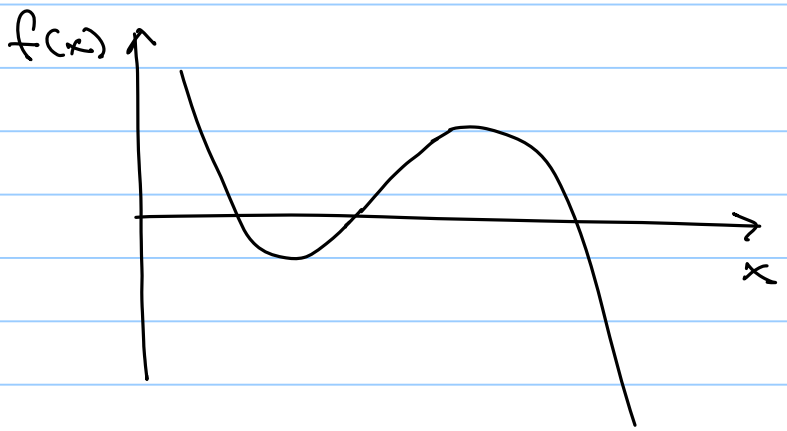


Fact: $x(t) \to \infty$ in FINITE TIME

Discuss re:
Edelstein-Keshet
M. Crichton
Quote!



Ex. 3 |



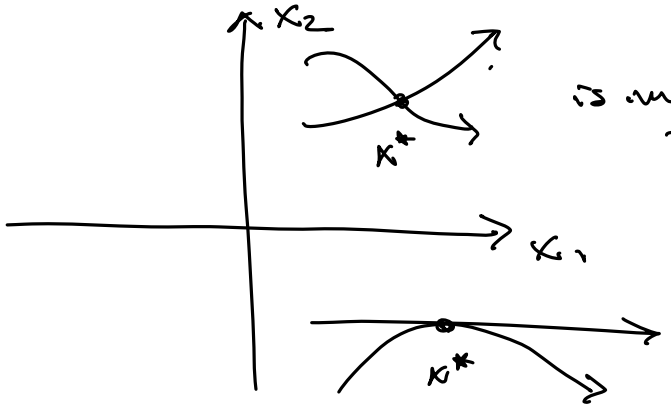
[sketch trajectories $x(t)$ on board ...]

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

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

Theorem: Let $\underset{\sim}{f}(x)$ 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:



is impossible: consider trajectories starting from x^*

Same story.

For strange examples when $f(x)$ is <u>Not</u> differentiable, try

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

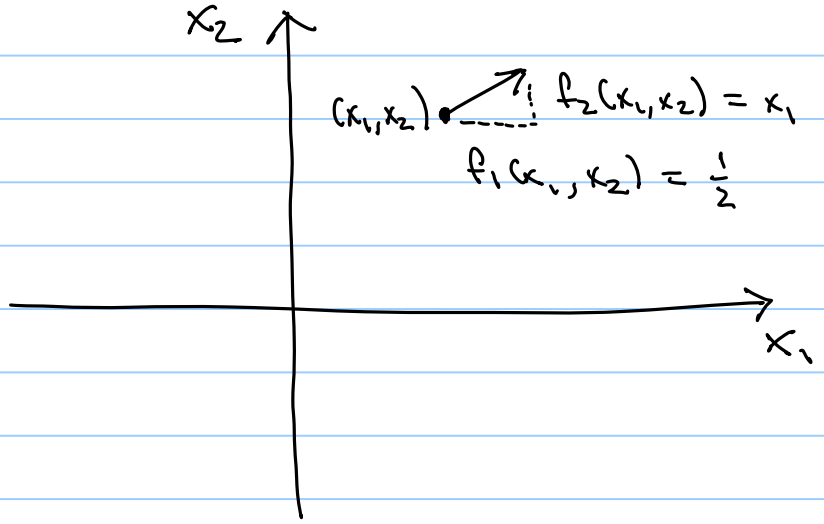
$x_\tau(t) = \begin{cases} 0 \\ \frac{1}{4}(t-\tau) \end{cases}$ is solution for any $\tau > 0$.

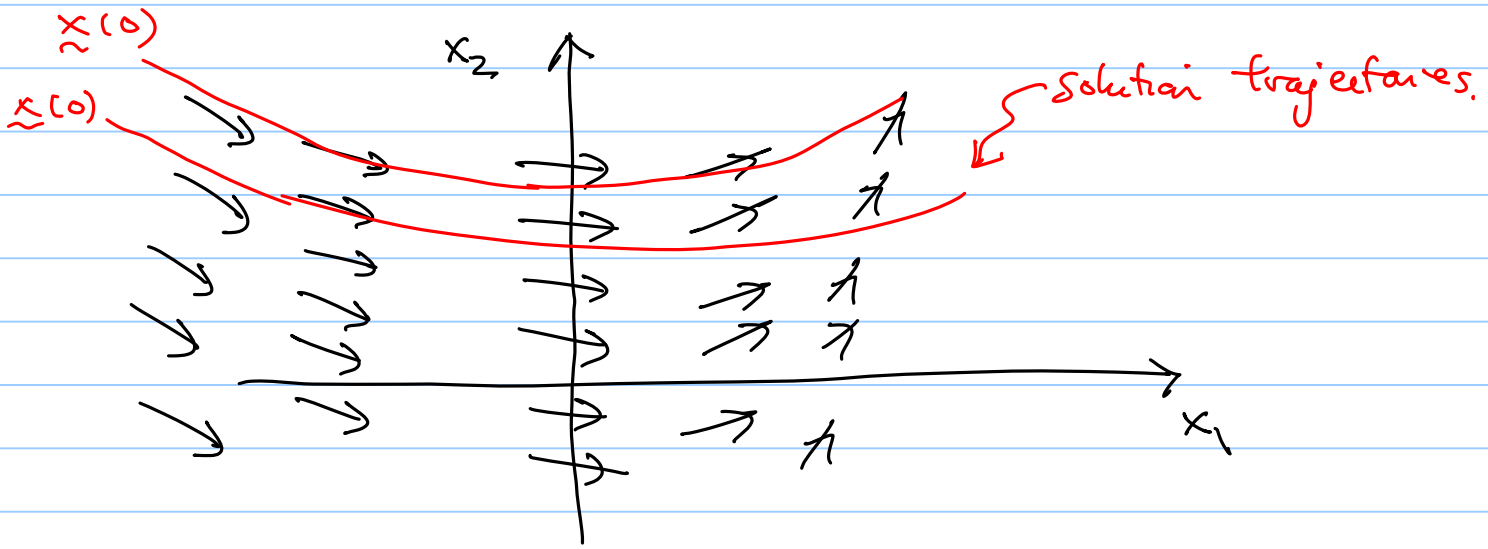
Visualize for $n = 2$ dimensions.

phase space = phase plane

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

Plot "direction field" [QUIVER PLOT]





- Dynamical systems studies trajectories that arise from following direction fields $\underset{\sim}{f}(\underset{\sim}{x})$.

Euler's Method...

$$\frac{dx_1}{dt} = f_1(x_1, x_2)$$

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

Say... at time t, 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

$$\rightarrow \quad x_1(t+\Delta t) = x_1(t) + f_1(x_1(t), x_2(t))\,\Delta t$$

DISTANCE = RATE × TIME

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

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

(RETURN TO DIR' FIELD PLOT, SCALE BY Δt)

(More on numerical timestepping schemes next time.)

2011, SKIP LORENZ; START COVERING AT ODEFUN

- When dimension $n > 2$, EXTREMELY complex and beautiful trajectories (dynamics) occur... for simple-looking differential equations

Ex 1)

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

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

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

Lorenz system.

MATLAB DEMO
lorenz.m

NOTE: Can also define: TIME-DEPENDENT velocity fields.

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

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 $\underset{\sim}{f}(\underset{\sim}{x})$
MATLAB: an "odefun"

describe system there!

- Suggested filename: lorenz_odefun.m

always include time

• Syntax.

function dxdt = lorenz_odefun (t, x)

x is a vector

x1 = x(1)

x2 = x(2)

x3 = x(t)

sigma = 10

rho = 28

beta = 8/3

dx1/dt = sigma*(x1 - x2)

dx2/dt = x1 * (rho - x3) - x2

dx3/dt = x1 * x2 - beta * x3

dxdt = [dx1/dt ;
dx2/dt ;
dx3/dt]

← RETURN A COLUMN VECTOR.

OR

function dxdt = simple_odefun (t, x)

[fill in: class]

x1 = x(1)

x2 = x(2)

dx1/dt = 1/2

dx2/dt = x1

dxdt = [dx1/dt ; dx2/dt]

(Assume $n=2$ dimensions)

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

"quiver plot"

See:

CODE direction_field_plotter_and_euler_method_simple.m
students develop similar code on set8 (so code not posted)

direction-field-plotter.m

Please note COMMENTS in this code, explaining:

- meshgrid
- quiver commands

3) Apply Euler's method → soln trajectory