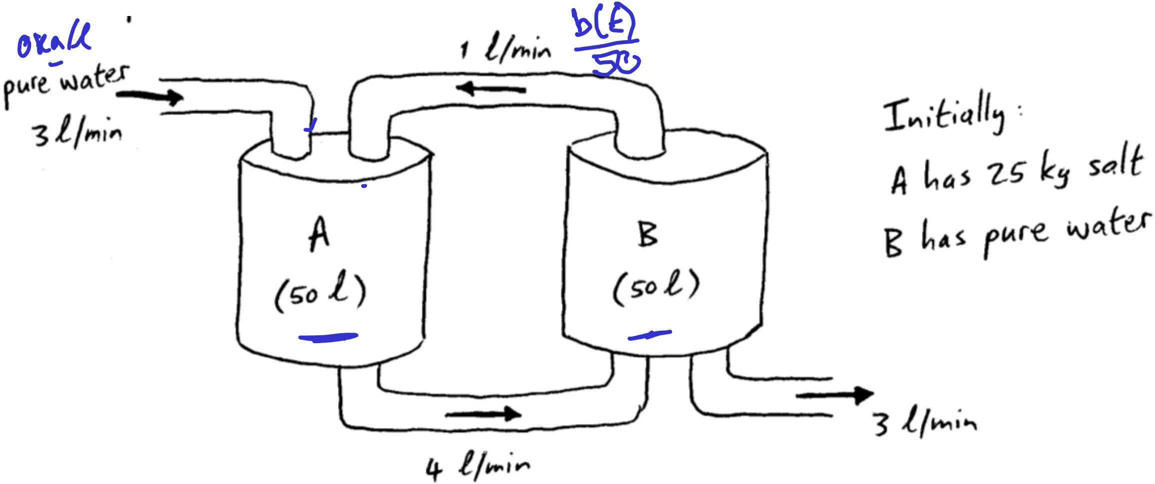
# Applied differential equations

## TW244 - Lecture 13

### 3.3: Systems of first-order DEs
### Example applications

Prof Nick Hale - 2020

Consider the following set-up (see p. 107 of Z&W):




Let $a(t)$ and $b(t)$ be the amount (kg) of salt in tanks $A$ and $B$, respectively.
As in Lecture 8, we assume that the salt is well-mixed in the fluid.

# 3.3: Systems of first-order DEs

## Application 2: System of tanks (cont.)

Note that the amount of liquid entering the tanks is the same as the amount leaving, so the volumes remain constant.

The rate of change in the amount of salt is given by:

$$\boxed{\boxed{\text{rate at which salt increases}} = \boxed{\text{rate at which salt flows in}} - \boxed{\text{rate at which salt flows out}}}$$

so that

$$\frac{da}{dt} = \cancel{(0\tfrac{kg}{\ell})(3\tfrac{\ell}{min})} + (\tfrac{b}{50}\tfrac{kg}{\ell})(1\tfrac{\ell}{min}) - (\tfrac{a}{50}\tfrac{kg}{\ell})(4\tfrac{\ell}{min}) = -\frac{2}{25}a + \frac{1}{50}b,$$

$$\frac{db}{dt} = (\tfrac{a}{50}\tfrac{kg}{\ell})(4\tfrac{\ell}{min}) - (\tfrac{b}{50}\tfrac{kg}{\ell})(1\tfrac{\ell}{min}) - (\tfrac{b}{50}\tfrac{kg}{\ell})(3\tfrac{\ell}{min}) = \frac{2}{25}a - \frac{2}{25}b.$$

So we have

$$\begin{aligned} \frac{da}{dt} &= -\frac{2}{25}a + \frac{1}{50}b, & a(0) &= 25 \\ \frac{db}{dt} &= \frac{2}{25}a - \frac{2}{25}b, & b(0) &= 0 \end{aligned}$$

or in matrix form:

$$\boxed{\frac{d}{dt}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} -\frac{2}{25} & \frac{1}{50} \\ \frac{2}{25} & -\frac{2}{25} \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix}.}$$

Solution...? We will look at the following techniques in this course:

- Elimination [Section 4.9]
- Eigenvalues and eigenvectors [Section 8.2]
- Numerically [End of this lecture & Assignment 03]

Let $x(t)$ be the population of predators (e.g., wolves) at time $t$.
Let $y(t)$ be the population of prey (e.g., rabbits) at time $t$.

We assume that the rabbits eat vegetation and the wolves eat rabbits.

- Without rabbits: $\frac{dx}{dt} = -ax$ (i.e., wolves decrease exponentially)
  With rabbits: $\frac{dx}{dt} = -ax + bxy$ (i.e., wolves may increase)
- Without wolves: $\frac{dy}{dt} = cy$ (i.e., rabbits increaes exponentially)
- With wolves: $\frac{dy}{dt} = cy - dxy$ (i.e., rabbits may decrease)

$\frac{dy}{dt} = ey \;;\; e = c - dx$

This gives the classic Lotka–Volterra model (with $a, b, c, d > 0$):

$$\boxed{\begin{aligned} \frac{dx}{dt} &= -ax + bxy \\ \frac{dy}{dt} &= cy - dxy. \end{aligned}}$$

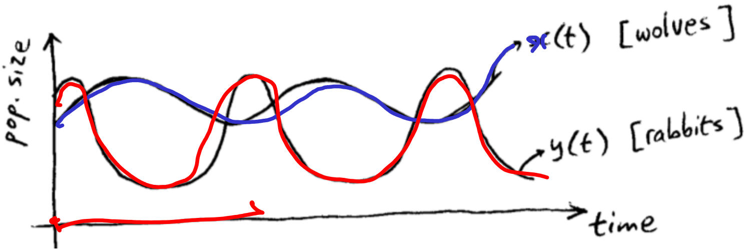
$, x(0) = x_0$
$, y(0) = y_0$

Why, for example $\frac{dy}{dt} = cy - dxy$? Suppose $x$ is fixed, then we have solution $y(t) = y_0 e^{(c-dx)t}$, thus we can think of the $-dxy$ term as limiting the exponential growth.

# 3.3: Systems of first-order DEs
## Application 3: Predator-prey models (cont.)

Unfortunately, for this system it is not possible to write the solution in closed form (i.e., in terms of elementary functions). We have to solve it numerically (~~see assignment 2!~~)

When we do plot the numerically computed solution, it typically looks like:




demonstrating a cyclic interaction between the predators and prey.

Such cycles can be seen in real data!

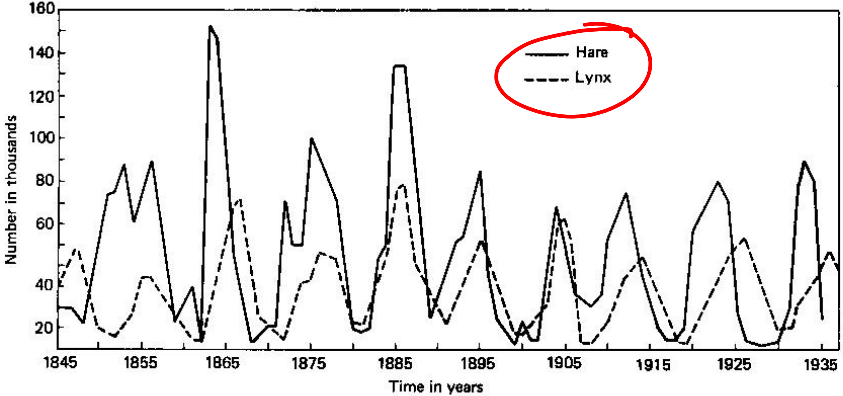



**Figure 9-3.** Changes in the abundance of the lynx and the snowshoe hare, as indicated by the number of pelts received by the Hudson's Bay Company. This is a classic case of cyclic oscillation in population density. (Redrawn from MacLulich 1937.)

Now suppose that two species compete for the same resources: $x = x(t)$ is the number of species 1 & $y = y(t)$ is the number of species 2. This gives rise to the model (with $a, b, c, d > 0$):

$$\begin{aligned}\frac{dx}{dt} &= ax - bxy,\\ \frac{dy}{dt} &= cy - dxy.\end{aligned}$$

Note how the signs differ from the Predator-Prey model. Here both species will decrease (at possibly different rates) due to intereaction.

The model can be extended to include logistic terms (recall Lecture 11):

$$\begin{aligned}\frac{dx}{dt} &= ax - bx^2 - cxy,\\ \frac{dy}{dt} &= dy - ey^2 - fxy.\end{aligned}$$

Later in the course we will learn to analyse such systems without solving them using phase diagrams. For now we consider numerical solutions.

`ode45` is a function in MATLAB that solves 1st-order initial value problems numerically. Read the document below or type `doc ode45` in MATLAB to learn more about its usage. For details of the algorithm, see TW324.

http://appliedmaths.sun.ac.za/TW244/files/ode45quickstart.pdf

`scipy.integrate.ode` is the Python equivalent of `ode45`. Feel free to use this instead.

Consider again the mixing tank example*

$$\begin{aligned} \frac{da}{dt} &= -\frac{2}{25}a + \frac{1}{50}b, \qquad & a(0) &= 25 \\ \frac{db}{dt} &= \frac{2}{25}a - \frac{2}{25}b, \qquad & b(0) &= 0. \end{aligned}$$

We vectorise, i.e., write $\underline{x} = \begin{pmatrix} a \\ b \end{pmatrix}$, and write a function $f$ so that $\frac{d\underline{x}}{dt} = f(t, \underline{x})$.

*Note, we shall see later that we can find an analytical solution to this DE so a numerical one isn't necessary. We do this here just for practice.

Let $x_1 = a$ and $x_2 = b$, then we have that

$$
\begin{aligned}
\frac{dx_1}{dt} &= f_1(t, x) = -\frac{2}{25}x_1 + \frac{1}{50}x_2, & x_1(0) &= 25 \\
\frac{dx_2}{dt} &= f_2(t, x) = \phantom{-}\frac{2}{25}x_1 - \frac{2}{25}x_2, & x_2(0) &= 0,
\end{aligned}
$$

or, in vector form

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$f(t, \underline{x}) = \begin{pmatrix} -\frac{2}{25}x_1 + \frac{1}{50}x_2 \\ \frac{2}{25}x_1 - \frac{2}{25}x_2 \end{pmatrix}, \qquad \underline{x}(0) = \begin{pmatrix} 25 \\ 0 \end{pmatrix}.$$

In MATLAB, we can write this using an *anonymous function*:

```
f  = @(t,x) [-2/25*x(1) + 1/50*x(2) ; 2/25*x(1) - 2/25*x(2)]
x0 = [25 ; 0]
```

Recall that this notation means “$f$ $\underbrace{\text{is a function of}}_{@}$ $t$ and $x$”.

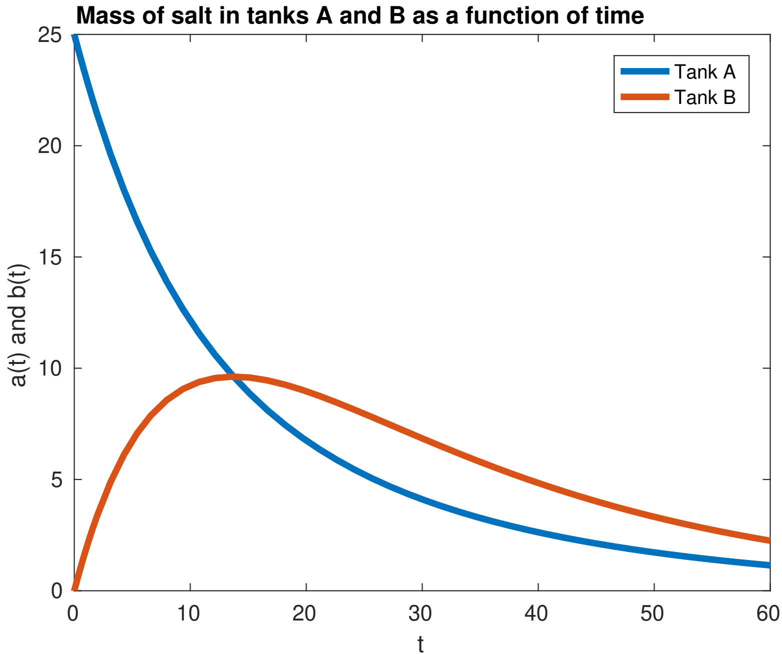
The calling sequence is `[TOUT,YOUT] = ode45(ODEFUN,TSPAN,Y0)`, where

- `ODEFUN` is our function $f$
- `Y0` is in this case $\underline{x}(0) = \begin{bmatrix} 25 \\ 0 \end{bmatrix}$
- `TSPAN = [T0 TFINAL]` is the time interval in which we want the solution.†

If we want the solution for say, the first 60 minutes, we would then call

```
f  = @(t, x) [-2/25*x(1) + 1/50*x(2) ; 2/25*x(1) - 2/25*x(2)];
x0 = [25 ; 0];                      % Initial concentrations.
tspan = [0, 60];                    % Run for one hour.
[t, x] = ode45(f, tspan, x0);      % Solve.
a = x(:,1); b = x(:,2);            % Solution components and plot:
plot(t, a, '-b', t, b, '-r', 'LineWidth', 3), shg
```

†It is also possible to give a TSPAN that is a vector of three or more points. Read doc ode45 for details.



Mass of salt in tanks A and B as a function of time
a(t) and b(t)
t
Tank A
Tank B
0
5
10
15
20
25
0
10
20
30
40
50
60


SIR