Systems of ODEs

Toy Example: Here is an extremely simplistic model of disease spread:

\[ \begin{align*}
H(t) &= \text{# healthy people at time } t \text{ (in years)} \\
I(t) &= \text{# infected people at time } t \\
D(t) &= \text{# dead people at time } t
\end{align*} \]

Assumptions:

1. Healthy people are infected at a rate of 
   \[ 0.3 \times \text{#healthy people} \]
2. Infected people recover at a rate of 
   \[ 0.9 \times \text{#infected people} \]
3. Infected people die at a rate of 
   \[ 0.1 \times \text{#infected people} \]

In equations:

1. \[ \frac{dH}{dt} = \text{infected} - 0.3H + 0.9I \]
2. \[ \frac{dI}{dt} = 0.3H - 0.9I - 0.1I \]
3. \[ \frac{dD}{dt} = 0.1I \]
Matrix Form: let \( \mathbf{u}(t) = (H(t), I(t), D(t)) \). 
\[
\frac{d\mathbf{u}(t)}{dt} = \mathbf{u}'(t) = \begin{bmatrix} -0.3 & 0.9 & 0 \\ 0.3 & -0.9 & 0.1 \\ 0 & 0.1 & 0 \end{bmatrix} \mathbf{u}(t)
\]

**Def:** A system of linear ordinary differential equations (ODEs) is a system of equations in unknown functions \( u_1(t), ..., u_n(t) \) equating the derivatives \( u_i' \) with a linear combination of the \( u_i \):
\[
\begin{align*}
    u_1'(t) &= a_{11}u_1(t) + \cdots + a_{1n}u_n(t) \\
    u_2'(t) &= a_{21}u_1(t) + \cdots + a_{2n}u_n(t) \\
    &\vdots \\
    u_n'(t) &= a_{n1}u_1(t) + \cdots + a_{nn}u_n(t)
\end{align*}
\]

Matrix form: writing \( \mathbf{u}(t) = (u_1(t), ..., u_n(t)) \) and \( \mathbf{u}'(t) = (u_1'(t), ..., u_n'(t)) \), a system of linear ODEs has the form
\[
\mathbf{u}'(t) = A\mathbf{u}(t)
\]
for an \( n \times n \) matrix \( A \)
(with numbers in it, not functions of \( t \)).

If you also specify the initial value \( \mathbf{u}(0) = \mathbf{u}_0 \), this is called an initial value problem.
How to solve a system of linear ODEs?

Diagonalize $A$!

Eg: Suppose $u_0$ is an eigenvector of $A$: $Au_0 = \lambda u_0$.

Then the solution of the initial value problem

$$u' = Au \quad u(0) = u_0$$

is

$$u(t) = e^{\lambda t} u_0$$

$$A u(t) = A e^{\lambda t} u_0 = e^{\lambda t} A u_0 = \lambda e^{\lambda t} u_0$$

$$u(0) = e^{0} u_0 = u_0$$

In general, we expand $u_0$ in an eigenbasis, as for difference equations:

$$u_0 = x_1 \omega_1 + \ldots + x_n \omega_n \quad A \omega_i = \lambda_i \omega_i$$

$$u(t) = e^{\lambda_1 t} x_1 \omega_1 + \ldots + e^{\lambda_n t} x_n \omega_n$$

is the solution of $u' = Au$, $u(0) = u_0$.

Check:

$$u'(t) = \lambda_1 e^{\lambda_1 t} x_1 \omega_1 + \ldots + \lambda_n e^{\lambda_n t} x_n \omega_n$$

$$A u(t) = e^{\lambda_1 t} x_1 A \omega_1 + \ldots + e^{\lambda_n t} x_n A \omega_n$$

$$= \lambda_1 e^{\lambda_1 t} x_1 \omega_1 + \ldots + \lambda_n e^{\lambda_n t} x_n \omega_n$$

$$u(0) = e^{0} x_1 \omega_1 + \ldots + e^{0} x_n \omega_n = u_0$$
Eg: In our infectious disease model, suppose
\[ n_0 = (1000, 1, 0) \] (1000 healthy people, 1 infected, 0 dead)

Eigenvalues of \( A = \begin{pmatrix} -0.3 & 1 & 0 \\ 0.3 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \) are

\[ \lambda_1 \approx -0.0235 \]
\[ \lambda_2 \approx -1.28 \]
\[ \lambda_3 = 0 \]

Eigenvalues are

\[ w_1 \approx \begin{pmatrix} 11.77 \\ -12.77 \\ 1 \end{pmatrix}, \quad w_2 \approx \begin{pmatrix} -0.765 \\ -255 \\ 1 \end{pmatrix}, \quad w_3 = (0) \]

Solve \( u_0 = x_1 w_1 + x_2 w_2 + x_3 w_3 \):

\[ u_0 = \begin{pmatrix} 1000 \\ 0 \\ 0 \end{pmatrix} \approx 18.70 w_1 - 1019.70 w_2 + 1001 w_3 \]

Solution is:

\[ u(t) = e^{-0.0235t} (18.70 w_1 - e^{-1.28t} \cdot 1019.70 w_2 + 1001 w_3) \]
\[ H(t) = 220 e^{-0.0235t} + 780 e^{-1.28t} \]
\[ I(t) = -238 e^{-0.0235t} + 239 e^{-1.28t} \]
\[ D(t) = 18.7 e^{-0.0235t} - 1019.7 e^{-1.28t} + 1001 \]

Looks like the human race is doomed...
Procedure for solving a linear system of ODEs using diagonalization:

To solve \( u' = Au, \quad u(0) = u_0 \) when \( A \) is diagonalizable:

1. Diagonalize \( A \): get an eigenbasis \( \{\omega_0, \ldots, \omega_n\} \) with eigenvalues \( \lambda_0, \ldots, \lambda_n \).
2. Expand \( u_0 \) in the eigenbasis:
   \[
   u_0 = x_0 \omega_0 + \cdots + x_n \omega_n
   \]

Solution:
\[
   u(t) = e^{\lambda_0 t} x_0 \omega_0 + \cdots + e^{\lambda_n t} x_n \omega_n
   \]

Compare to:

Procedure for solving a Difference Equation using diagonalization:

To solve \( v_{k+1} = Av_k, \quad v_0 \) fixed when \( A \) is diagonalizable:

1. Diagonalize \( A \): get an eigenbasis \( \{\omega_0, \ldots, \omega_n\} \) with eigenvalues \( \lambda_0, \ldots, \lambda_n \).
2. Expand \( v_0 \) in the eigenbasis:
   \[
   v_0 = x_0 \omega_0 + \cdots + x_n \omega_n
   \]

Solution:
\[
   v_k = \lambda^{k}_0 x_0 \omega_0 + \cdots + \lambda^{k}_n x_n \omega_n
   \]
This works fine with complex eigenvalues. As with difference equations, you can write the solution with real numbers using trig functions.

E.g.: \( u_1'(t) = u_2, \quad u_2'(t) = -4u_1, \)
\[ u_1(0) = 2 \quad u_2(0) = 0 \]
\[ \implies \mathbf{u}' = A \mathbf{u} \quad \text{for} \quad A = \begin{pmatrix} 0 & 1 \\ -4 & 0 \end{pmatrix} \quad \mathbf{u}(0) = \begin{pmatrix} 2 \\ 0 \end{pmatrix} \]

Eigenvalues are \( \lambda = 2i, \quad \bar{\lambda} = -2i \)

Eigenvectors are \( \mathbf{w} = \begin{pmatrix} 1 \\ 2i \end{pmatrix} \quad \bar{\mathbf{w}} = \begin{pmatrix} 1 \\ -2i \end{pmatrix} \)

Solve \( \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} = x_1 \mathbf{u} + x_2 \bar{\mathbf{w}} \quad \implies x_1 = x_2 = 1 \)

Solution is \( \mathbf{u}(t) = e^{\lambda t} \mathbf{w} + e^{\bar{\lambda} t} \bar{\mathbf{w}} = 2 \Re \left[ e^{\lambda t} \mathbf{w} \right] \)
\[ = 2 \Re \left[ e^{2i t} \begin{pmatrix} 2i \\ 1 \end{pmatrix} \right] = 2 \Re \left[ (\cos(2t) + i \sin(2t)) \begin{pmatrix} 1 \\ 2i \end{pmatrix} \right] \]
\[ = 2 \Re \left( \begin{pmatrix} \cos(2t) + i \sin(2t) \\ -2i \sin(2t) + 2 \cos(2t) \end{pmatrix} \right) = \begin{pmatrix} 2 \cos(2t) \\ -4 \sin(2t) \end{pmatrix} \]

Check: \( u_1' = (2 \cos(2t))' = -4 \sin(2t) = u_2 \)
\( u_2' = (-4 \sin(2t))' = -8 \cos(2t) = -4u_1 \)
\( u_1(0) = 2 \quad u_2(0) = 0 \)
This method can also be used to solve (linear) ODEs containing higher-order derivatives.

**Eg:** Hooke’s Law says the force applied by a spring is proportional to the amount it is stretched or compressed:

\[ F(t) = -k \ p(t) \quad k \geq 0 \]

\[ F = ma, \quad a = \text{acceleration} = p'' \text{; replace } k \text{ by } k/m: \]

\[ p''(t) = -k p(t) \]

**Trick:** Let \( u_1 = p, \ u_2 = p' \). Then

\[ u_1' = u_2, \quad u_2' = -ku_1. \]

This is the system

\[ u'(t) = \begin{pmatrix} 0 & 1 \\ -k & 0 \end{pmatrix} u(t). \]

We solved this before for \( k=4 \), \( u(0) = (2,0) \):

\[ p(t) = 2 \cos(2t) \]

\[ p'(t) = -4 \sin(2t) \quad \text{oscillation.} \]
The Matrix Exponential

There are 2 features missing from the ODEs picture that we had for difference equations:

1. Matrix form: \( V_k = C D^k C^{-1} V_0 \)

2. Existence of solutions:
   - it's obvious that \( V_k = A^k V_0 \) has a solution
   - it was not obvious how to compute it.

Both can be filled in using the matrix exponential.

Recall: Using Taylor expansions, you can write
\[
e^x = 1 + x + \frac{1}{2!} x^2 + \frac{1}{3!} x^3 + \ldots \quad \text{(convergent sum)}
\]

Def: Let \( A \) be an \( n \times n \) matrix. The matrix exponential is the \( n \times n \) matrix
\[
e^A = I_n + A + \frac{1}{2!} A^2 + \frac{1}{3!} A^3 + \ldots \quad \text{(convergent sum)}
\]

Eg: \( A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \) \( \Rightarrow A^2 = 0 \), so
\[
e^{At} = I_2 + At + 0 + \ldots = \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix}
\]
Eg: \( A = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda^2 \end{pmatrix} \Rightarrow A^k = \begin{pmatrix} \lambda^k & 0 \\ 0 & \lambda^{2k} \end{pmatrix} \), so
\[
e^{At} = \begin{pmatrix} e^{\lambda t} & 0 \\ 0 & e^{\lambda^2 t} \end{pmatrix}
\]Why do we care about \( e^{At} \)?

Fact: \( \frac{d}{dt} e^{At} = Ae^{At} \)

Consequence: \( u(t) = e^{At} u_0 \) solves the linear ODE
\[
u(t) = Au(t) \quad u(0) = u_0
\]
In particular, a solution exists.

The equations
\[
u(t) = e^{At} u_0 \quad \text{and} \quad \nu_k = A^k \nu_0
\]
are analogous: they both show a solution exists, but give you no way to compute it.

Eg: If \( A = CDC^{-1} \) is diagonalizable then
\[
e^{At} = Ce^{Dt} C^{-1} = C \begin{pmatrix} e^{\lambda_1 t} & 0 \\ 0 & e^{\lambda_2 t} \end{pmatrix} C^{-1}
\]This is computable!
The equations
\[ e^{At} = C e^{D} C^{-1} \] and \[ A^k = C D^k C^{-1} \]
are also analogous; they are computable!

In fact, if you expand out
\[ u(t) = C e^{D} C^{-1} u_0 \]
you exactly get the vector form
\[ u(t) = e^{\lambda_1 t} x_1 \omega_1 + \ldots + e^{\lambda_n t} x_n \omega_n \]
where \( \left( x_1, \ldots, x_n \right) = C^{-1} u_0 \).

<table>
<thead>
<tr>
<th>Difference Equation</th>
<th>Dictionary</th>
<th>Initial Value Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_{k+1} = A V_k ) ( v_0 ) fixed</td>
<td>Problem ( u'(t) = A u(t) ) ( u(0) ) fixed</td>
<td></td>
</tr>
<tr>
<td>( V_k = A^k v_0 )</td>
<td>Uncountable Solution</td>
<td>( u(t) = e^{At} u(0) )</td>
</tr>
<tr>
<td>( V_k = \lambda_1^{k} x_1 \omega_1 + \ldots + \lambda_n^{k} x_n \omega_n )</td>
<td>Computable Solution for ( V_0 = x_1 \omega_1 + \ldots + x_n \omega_n ) ( \text{(clen diagonalizable)} )</td>
<td></td>
</tr>
</tbody>
</table>

\[ A^k = C D^k C^{-1} \] Matrix Form \[ e^{At} = C e^{D} C^{-1} \]