Lecture 24: Systems of Ordinary Differential Equations

Reading:
Kreyszig Sections: 4.1, 4.2 (pages131–135, 136–139)

Systems of Ordinary Differential Equations

The ordinary differential equations that have been treated thus far are relations between a single function and how it changes:

$$F(\frac{d^n y}{dx^n}, \frac{d^{n-1} y}{dx^{n-1}}, \ldots, \frac{dy}{dx}, y, x) = 0 \tag{24-1}$$

Many physical models of systems result in differential relations between *several* functions. For example, a first-order system of ordinary differential equations for the functions $(y_1(x), y_2(x), \ldots, y_n(x))$ is:

$$\begin{aligned} \frac{dy_1}{dx} &= f_1(y_1(x), y_2(x), \ldots, y_n(x), x) \\ \frac{dy_2}{dx} &= f_2(y_1(x), y_2(x), \ldots, y_n(x), x) \\ \vdots &= \vdots \\ \frac{dy_n}{dx} &= f_n(y_1(x), y_2(x), \ldots, y_n(x), x) \end{aligned} \tag{24-2}$$

or with a vector notation,

$$\frac{d\vec{y}(x)}{dx} = \vec{f}(\vec{y}, x) \tag{24-3}$$

Example: The Spread of a MIT Joke

The predator-prey model serves as the classical example of a system of differential equations. This is a (possibly humorous) variant of the predator-prey problem.

Suppose there is a fairly bad joke that circulates around the student population. Students either know the joke or they don't and thus can be divided into two populations:

Jaded, J Knows the joke, and if someone tries to tell it to them, they interrupt with, "Yeah, Yeah. I heard that one. It's pretty, like, stupid."

Naive, N Never heard the joke or has forgotten it.

As the joke spreads, or as students graduate, or students forget the joke, or as new students are admitted to MIT, the populations change.

We will try to construct a model that reflects how the populations change each day.

We will suppose that freshman enter MIT a constant daily rate; in order to keep the population of students regulated, the admissions office accepts freshman at a rate that depends on how many of 4000 slots are open. Therefore, freshman enter MIT, and thus the Naive population at a daily rate of:

$$\frac{dN_{frsh}}{dt} = \frac{4000 - (J + N)}{365} \tag{24-4}$$

Students have a lot of things on their mind (some of which is education) and so they tend to be forgetful. Students who know the joke tend to forget at rate ϕ/year. Suppose that a fraction, ϕ, of the Jaded students forget the joke each year—these students leave the J-group and enter the N-group at a daily rate:

$$\begin{aligned}\frac{dJ_{forg}}{dt} &= \frac{\alpha_A J}{365} = \alpha J \\ \frac{dN_{forg}}{dt} &= -\frac{\alpha_A J}{365} = -\alpha J\end{aligned} \tag{24-5}$$

It is closely held secret that Susan Hockfield, MIT's president, has an odd sense of humor. At each commencement ceremony, as the proud candidates for graduation approach the president to collect their hard-earned diploma, President Hockfield whispers to the student, "Have you the joke about…?" If the student says, "Yes. I have heard that joke. It is *very* funny!!!" then the diploma is awarded. However, if the student says, "No. But, I am dying to hear it!!!", the president's face drops into a sad frown and the student is asked to leave without collecting the diploma.[16]

Therefore, only students in the J-group can graduate. Let's assume that at any one time, 1/3 of the jaded students have satisfied the graduation requirements, and of this group 99% will graduate:

$$\frac{dJ_{grdt}}{dt} = -\frac{0.99J/3}{365} = -\gamma J \tag{24-6}$$

The joke spreads in proportion to its "funniness coefficient" and the probability that a naive student runs into a jaded student:

$$\begin{aligned}\frac{dJ_{sprd}}{dt} &= -\frac{\phi_A JN}{365^2} = -\phi JN \\ \frac{dN_{sprd}}{dt} &= \frac{\phi_A JN}{365^2} = \phi JN\end{aligned} \tag{24-7}$$

Therefore, an iterative model for the student population that knows the joke is:

$$\begin{aligned}\text{Naive Fraction(Tomorrow)} &= \text{Naive Fraction(Today)} + \text{Change in Naive Fraction} \\ \text{Jaded Fraction(Tomorrow)} &= \text{Jaded Fraction(Today)} + \text{Change in Jaded Fraction}\end{aligned} \tag{24-8}$$

or

$$\begin{aligned}N_{i+1} &= N_i + \frac{4000 - (N_i + J_i)}{365} + \alpha J_i - \phi J_i N_i \\ J_{i+1} &= J_i + \phi J_i N_i - \gamma J_i - \alpha J_i\end{aligned} \tag{24-9}$$

[16] This event is an annual source of confusion and embarrassment for the students' proud families—and a source of sadistic amusement to the attending faculty (who have an even stranger sense of humor than Hockfield's).

Lecture 24 MATHEMATICA® Example 1

Iterative Example of Predator-Prey Simulation

Download notebooks, pdfs, or html from http://pruffle.mit.edu/3.016-2006.

Functions to the simulate system of iterative equations, Eq. 24-8, are developed.

1: *FreshmanEntranceRate* takes the current Jaded and Naive populations and returns how many students are admitted to the Naive popuation each day.

2: The number of students that forget the joke each day is proportional to the current Jaded population.

3: Because only Jaded students can graduate, *GraduationRate* is proportional to the current Jaded population.

4: *JokeSpreadRate* depends linearly on the probability that a Jaded meets a Naive student, and therefore the spread rate depends on the product of the two populations.

6: *TomorrowsNaive* and *TomorrowsJaded* advance the two populations by one day.

7: If the two populations are kept in a two-item list, then *TomorrowsPopulation* will advance the entire population list.

```
1 FreshmanEntranceRate[Naive_, Jaded_] :=
    (4000 - (Jaded + Naive))/365;
```

About half the people who know the joke forget it each year and only the Jaded know the joke. The model for how many forget the joke each day is:

```
2 PopulationForgetfullness = .5;
  ForgotJoke[Naive_, Jaded_] :=
    PopulationForgetfullness * Jaded/365
```

The model for how many leave each day by graduating is:

```
3 GraduationCoefficient = 0.99^(1/3);
  GraduationRate[TodaysNaive_, TodaysJaded_] :=
    GraduationCoefficient * TodaysJaded/365;
```

The rate that the joke spreads will determine how many of the Naive will become Jaded. The probability that a Naive meets a Jaded who tells the joke is proportional to the joke funniness and the daily probability that the two meet.

```
4 JokeFunniness = 0.35;
  JokeSpreadRate[Naive_, Jaded_] :=
    JokeFunniness * Naive * Jaded/(365 * 365)
```

```
5 TomorrowsNaive[TodaysNaive_, TodaysJaded_] :=
    TodaysNaive +
      FreshmanEntranceRate[TodaysNaive, TodaysJaded] -
      JokeSpreadRate[TodaysNaive, TodaysJaded] +
      ForgotJoke[TodaysNaive, TodaysJaded]
```

```
6 TomorrowsJaded[TodaysNaive_, TodaysJaded_] :=
    TodaysJaded +
      JokeSpreadRate[TodaysNaive, TodaysJaded] -
      ForgotJoke[TodaysNaive, TodaysJaded] -
      GraduationRate[TodaysNaive, TodaysJaded]
```

```
7 TomorrowsPopulation[{TodaysNaive_, TodaysJaded_}] :=
    {TomorrowsNaive[TodaysNaive, TodaysJaded],
      TomorrowsJaded[TodaysNaive, TodaysJaded]}
```

Lecture 24 MATHEMATICA® Example 2

Visualizing the Spread of Jokes at MIT

Download notebooks, pdfs, or html from http://pruffle.mit.edu/3.016-2006.

The behavior of the iterative model presented in the above example are visualized.

```
1 NestList[TomorrowsPopulation, {400, 600}, 5]

For an animation of a population of {Naive, Jaded} = {0,0}

2 Table[ListPlot[NestList[TomorrowsPopulation, {0, 0}, i],
    PlotJoined → True, PlotStyle → {Hue[1], Thickness[0.01]},
    AxesLabel → {"Naive", "Jaded"},
    PlotRange –> {{0, 4000}, {0, 4000}}], {i, 10, 2500, 50}]

From the above animation, one might conclude that the
population will stably climb toware {4000,0}. In the following
animation, an initial population of {1,1} shows that a small
pertubation away from having "no jaded students to tell the joke"
has very different long-term behavior

3 Table[ListPlot[NestList[TomorrowsPopulation, {1, 1}, i],
    PlotRange –> {{0, 4000}, {0, 4000}}, PlotJoined → True,
    PlotStyle → {Hue[1], Thickness[0.01]},
    AxesLabel → {"Naive", "Jaded"}], {i, 10, 2500, 50}]

Now calculate trajectories for a variety of initial conditions for the
jaded and naive populations, selected randomly, then plot them
on the naive–jaded plane:

4 graphicslist = Table[ListPlot[NestList[TomorrowsPopulation,
      {4000 * Random[], 4000 * Random[]}, 8000],
    PlotRange → All, PlotJoined → True,
    PlotStyle → {RGBColor[Random[],
        Random[], Random[]], Thickness[0.005]},
    DisplayFunction → Identity], {i, 1, 50}]

5 Show[graphicslist, DisplayFunction → $DisplayFunction,
  AxesLabel → {"Naive", "Jaded"},
  PlotRange –> {{0, 4000}, {0, 8000}}]

The trajectories' convergence point calculated numerically:

6 fp = FixedPoint[TomorrowsPopulation, {800, 2300}, 20000]

7 winsize = 10^(–8);
  plotrange = {{fp[[1]] – winsize, fp[[1]] + winsize},
    {fp[[2]] – winsize, fp[[2]] + winsize}}
  Show[graphicslist, DisplayFunction → $DisplayFunction,
    AxesLabel → {"Naive", "Jaded"}, PlotRange –> plotrange]
```

1: Here, `NestList` is used with the *TomorrowsPopulation* to create a sequence of population-lists, starting at $\{N, J\} = \{400, 600\}$, for five consectutive days.

2: The results of `NestList` are repeatly plotted by using `Table` at 50 day intervals. This will create sequence of plots that can be collected into an animation. In this case, an initial population of $\{0, 0\}$ will asymptotically approach $\{4000, 0\}$; at very long times, no one would graduate, one one would enter MIT, and no one would know the joke. This is not only bad policy, but the system is unstable...

3: In a similar animation to the above, any small positive perturbation from the $\{0, 0\}$ population will initially be driven towards the $\{4000, 0\}$ solution, but will 'veer' away and then rush towards another asymptotic solution.

4: To examine the behavior for a variety of initial populations, iteration sequences are generated for 50 different initial random populations, and then represented with an (undisplayed) graphics object. Each trajectory is identified with a random color.

6: The graphics example above indicates that trajectories tend towards a stable fixed point. The numerical value of the fixed point can be determined by iterating the list until the result ceases to change numerically. This method is implemented in the function `FixedPoint` which is insensitive to the initial population.

7: Here, the fixed point is used to create a small 'window' in `PlotRange` so that the local behavior can be observed.

The stability and behavior of the iterative model 24-8 can be analyzed by replacing the coupled iterative equations with coupled ODEs:

$$\begin{aligned} \frac{dN}{dt} &= \frac{4000 - (N + J)}{365} + \alpha J - \phi J N \\ \frac{dJ}{dt} &= \phi J N - \gamma J - \alpha J \end{aligned} \tag{24-10}$$

A *critical point* is one at which the left-hand side of Equations 24-2, 24-3, or 24-10 vanish—in other words, a critical point is a special value of the vector $\vec{y}(x)$ where the system of equations does not evolve. For the system defined by Eq. 24-10, there are two critical points

$$(N_{\text{unst}} = 4000,\ J_{\text{unst}} = 0) \qquad \text{and} \qquad \left(N_{\text{stab}} = \frac{\alpha + \gamma}{\phi},\ J_{\text{stab}} = \frac{4000\phi - \gamma - \alpha}{\phi + 365\gamma\phi}\right) \tag{24-11}$$

However, while a system that is sitting *exactly* at a critical point will not evolve, some of the critical points are not *stable*. There are three broad categories of critical points:[17]

[17] There are others that will be discussed later.

Stable Any slight perturbation of the system away from the critical point results in an evolution back to that critical point. In other words, all points in the neighborhood of a stable critical point have a trajectory that is attracted back to that point.

Unstable Some slight perturbation of the system away from the critical point results in an evolution away from that critical point. In other words, some points in the neighborhood of an unstable critical point have trajectories that are repelled by the point.

Circles Any slight perturbation away from a critical point results in an evolution that always remains near the critical point. In other words, all points in the neighborhood of a circle critical point have trajectories that remain in the neighborhood of the point.

Reduction of Higher Order ODEs to a System of First Order ODEs

Higher-order ordinary differential equations can usually be re-written as a *system* of first-order differential equations. If the higher-order ODE can be solved for its largest derivative:

$$\frac{d^n y}{dt^n} = F(\frac{d^{n-1}y}{dt^{n-1}}, \frac{d^{n-2}y}{dt^{n-2}}, \ldots, \frac{dy}{dt}, t) \tag{24-12}$$

then $n-1$ "new" functions can be introduced via

$$\begin{aligned}
y_0(t) &\equiv y(t) \\
y_1(t) &\equiv \frac{dy}{dt} = \frac{dy_0}{dt} \\
y_2(t) &\equiv \frac{d^2y}{dt^2} = \frac{dy_1}{dt} \\
&\vdots \equiv \vdots \\
y_{n-1}(t) &\equiv \frac{d^{n-1}y}{dt^{n-1}} = \frac{dy_{n-2}}{dt} \\
y_n(t) &\equiv \frac{d^n y}{dt^n} = F(\frac{d^{n-1}y}{dt^{n-1}}, \frac{d^{n-2}y}{dt^{n-2}}, \ldots, \frac{dy}{dt}, t) = \frac{dy_{n-1}}{dt}
\end{aligned} \tag{24-13}$$

or

$$\frac{d}{dt}\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-2} \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ F(y_{n-1}, y_{n-2}, \ldots, y_1, y_0, t) \end{pmatrix} \tag{24-14}$$

For example, the damped harmonic oscillator, $M\ddot{y}+\eta l_o\dot{y}+K_s y = 0$, can be re-written by introducing the momentum variable, $p = Mv = M\dot{y}$, as the system:

$$\begin{aligned}
\frac{dy}{dt} &= \frac{p}{M} \\
\frac{dp}{dt} &= -K_s y - \eta l_o p
\end{aligned} \tag{24-15}$$

which has only one critical point $y = p = 0$.

The equation for a free pendulum, $MR^2\ddot{\theta} + MgR\sin(\theta) = 0$, can be re-written by introducing the angular momentum variable, $\omega = MR\dot{\theta}$ as the system,

$$\begin{aligned} \frac{d\theta}{dt} &= \frac{\omega}{MR} \\ \frac{d\omega}{dt} &= -Mg\sin(\theta) \end{aligned} \tag{24-16}$$

which has two *different* kinds of critical points: $(\omega = 0, \theta = n_{even}\pi)$ and $(\omega = 0, \theta = n_{odd}\pi)$.

Finally, the beam equation $EI\frac{d^4y}{dx^4} = w(x)$ can be rewritten as the system:

$$\frac{d}{dx}\begin{pmatrix} y \\ m_{slope} \\ M \\ S \end{pmatrix} = \begin{pmatrix} m_{slope} \\ \frac{M}{EI} \\ S \\ w(x) \end{pmatrix} \tag{24-17}$$

where m_{slope} is the slope of the beam, M is the local bending moment in the beam, S is the local shearing force in the beam, and $w(x)$ is the load density.

This beam equation does not have any interesting critical points.

Linearization of Systems of ODEs

The fixed point plays a very important role in understanding the behavior of non-linear ODEs.

The general autonomous non-linear ODE can be written as:

$$\frac{d\vec{y}}{dt} \equiv \frac{d}{dt}\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} F_1(y_1, y_2, \ldots, y_n) \\ F_2(y_1, y_2, \ldots, y_n) \\ \vdots \\ F_n(y_1, y_2, \ldots, y_n) \end{pmatrix} \equiv \vec{F}(y_1, y_2, \ldots, y_n) \tag{24-18}$$

The fixed points are the solutions to:

$$\vec{F}(y_1^f, y_2^f, \ldots, y_n^f) = \begin{pmatrix} F_1(y_1^f, y_2^f, \ldots, y_n^f) \\ F_2(y_1^f, y_2^f, \ldots, y_n^f) \\ \vdots \\ F_n(y_1^f, y_2^f, \ldots, y_n^f) \end{pmatrix} = \vec{0} \tag{24-19}$$

If the fixed points can be found, then the behavior *near* the fixed points can be analyzed by linearization. Letting $\vec{\delta} = \vec{y} - \vec{y}^f$ be a point near a fixed point, then a linear approximation is:

$$\frac{d}{dt}\vec{\delta} = \underline{J}\delta \tag{24-20}$$

where

$$\underline{J} = \begin{pmatrix} \frac{\partial F_1}{\partial y_1}\Big|_{\vec{y}^f} & \frac{\partial F_2}{\partial y_1}\Big|_{\vec{y}^f} & \cdots & \frac{\partial F_n}{\partial y_1}\Big|_{\vec{y}^f} \\ \frac{\partial F_1}{\partial y_2}\Big|_{\vec{y}^f} & \frac{\partial F_2}{\partial y_2}\Big|_{\vec{y}^f} & \cdots & \frac{\partial F_n}{\partial y_2}\Big|_{\vec{y}^f} \\ \frac{\partial F_1}{\partial y_3}\Big|_{\vec{y}^f} & \frac{\partial F_2}{\partial y_3}\Big|_{\vec{y}^f} & \ddots & \vdots \\ \frac{\partial F_1}{\partial y_n}\Big|_{\vec{y}^f} & \cdots & \cdots & \frac{\partial F_n}{\partial y_n}\Big|_{\vec{y}^f} \end{pmatrix} \tag{24-21}$$

Equation 24-20 looks very much like a simple linear first-order ODE. The expression

$$\vec{y}(t) = e^{\underline{J}t}\vec{y}(t=0) \tag{24-22}$$

might solve it if the proper analog to the exponential of a matrix were known.

Rather than solve the matrix equation directly, it makes more sense to transform the system into one that is diagonalized. In other words, instead of solving Eq. 24-20 with Eq. 24-21 near the fixed point, find the eigenvalues, λ_i, of Eq. 24-21 and solve the simpler system by transforming the $\vec{\delta}$ into the eigenframe $\vec{\eta}$:

$$\begin{aligned} \frac{d\eta_1}{dt} &= \lambda_1\eta_1 \\ \frac{d\eta_2}{dt} &= \lambda_2\eta_2 \\ \vdots &= \vdots \\ \frac{d\eta_n}{dt} &= \lambda_n\eta_n \end{aligned} \tag{24-23}$$

for which solutions can be written down immediately:

$$\begin{aligned} \eta_1(t) &= \eta_1(t=0)e^{\lambda_1 t} \\ \eta_2(t) &= \eta_2(t=0)e^{\lambda_2 t} \\ \vdots &= \vdots \\ \eta_n(t) &= \eta_n(t=0)e^{\lambda_n t} \end{aligned} \tag{24-24}$$

If any of the eigenvalues of $\underline{J}$ have a positive real part positive, then an initial condition near that fixed point will diverge from that point—stability occurs only if all the eigenvalues are negative.

Lecture 24 MATHEMATICA® Example 3

Analyzing the Stability for the MIT Joke

Download notebooks, pdfs, or html from http://pruffle.mit.edu/3.016-2006.

It is shown that the MIT has an unstable and a stable fixed point for the parameters utilized in Example 24-1.

1: These expressions are the right-hand-sides of Eqs. 24-10.

2: A fixed point appears whenever the right-hand-sides of a system of coupled ODEs vanish. In this case, there will be two solutions defined in the list associated with *fixedpoint* .

3: *MITModel* will be defined as a short-hand for the replacement rule associated with the parameters used in Example 24-1.

4: Therefore, for the ODE model, this will calculate fixed points.

5: This is the Jacobian (i.e., Eq. 24-21) for Eq. 24-10.

6: The Jacobian analyzed at the fixed points, *fixedpoint* , will produce the characteristic matrix for the stability of each point.

8: A list is created with the `Eigenvalues` associated with the Jacobian at each fixed point.

9: The MIT model shows that there are two fixed points. The one with negative real parts of its eigenvalues is the stable solution and it has a slow oscillatory part.

The Joke-Model as the system of ODEs:

$$\frac{dN}{dt} = (4000 - (N + J))/365 - \phi\, J\, N + \alpha\, J$$

$$\frac{dJ}{dt} = \phi\, N\, J - \gamma\, J - \alpha\, J$$

Find the fixed points:

```
1 Ndot = (4000 - (N + J))/365 - ϕ J N + α J
  Jdot = ϕ J N - γ J - α J
2 fixedpoint = Solve[{Ndot == 0, Jdot == 0}, {N, J}]
3 MITModel = {ϕ -> 0.75/(365*365),
      α -> 0.5/365, γ -> 0.99(1/3)/365};
4 fixedpoint /. MITModel
```

Calculate the jacobian:

```
5 Jacob = {
      {D[Ndot, N], D[Ndot, J]},
      {D[Jdot, N], D[Jdot, J]}
    };
  Jacob // MatrixForm
```

Find the Jacobian at the fixed points:

```
6 JacobFixedPoints = Simplify[Jacob /. fixedpoint];
  JacobFixedPoints[[1]] // MatrixForm
7 JacobFixedPoints[[2]] // MatrixForm
```

Compute eigenvalues of the Jacobian at the fixed point

```
8 evals = {Eigenvalues[JacobFixedPoints[[1]]],
    Eigenvalues[JacobFixedPoints[[2]]]}
9 MITModel = {ϕ -> 0.35/(365*365),
      α -> 0.5/365, γ -> 0.99(1/3)/365};
  fixedpoint /. MITModel
  evals /. MITModel
```