§9.1 Systems of ODEs

Recall the regular ODE problem: find some $x(t)$ such that

$$\begin{cases} \frac{dx(t)}{dt} = f\left(t, x(t)\right), \\ x(a) = c, \end{cases}$$

where f, a, c are given.

Sometimes the physical systems we are considering are more complex. For example, we might be interested in the system of ODEs:

$$\begin{cases} \frac{dx(t)}{dt} = f\left(t, x(t), y(t)\right), \\ \frac{dy(t)}{dt} = g\left(t, x(t), y(t)\right), \\ x(a) = c, \\ y(a) = d. \end{cases}$$

Example 1. Consider the following system of ODEs:

$$\begin{cases} \frac{dx(t)}{dt} = t^2 - x \\ \frac{dy(t)}{dt} = y^2 + y - t, \\ x(0) = 1, \\ y(0) = 0. \end{cases}$$

You should immediately notice that this is not a system at all, but rather a collection of two ODEs:

$$\begin{cases} \frac{dx(t)}{dt} = t^2 - x \\ x(0) = 1, \end{cases}$$

and

$$\begin{cases} \frac{dy(t)}{dt} = y^2 + y - t, \\ y(0) = 0. \end{cases}$$

These two ODEs can be solved separately. We call such a system *uncoupled*. In an uncoupled system the function $f(t, x, y)$ is independent of y, and $g(t, x, y)$ is independent of x. A system which is not uncoupled, is, of course, *coupled*. We will not consider uncoupled systems.

Larger Systems

There is no need to stop at two functions. We may imagine we have to solve the following problem: find $x_1(t), x_2(t), \ldots, x_n(t)$ such that

$$\begin{cases} \frac{dx_1(t)}{dt} = f_1\left(t, x_1(t), x_2(t), \ldots, x_n(t)\right), \\ \frac{dx_2(t)}{dt} = f_2\left(t, x_1(t), x_2(t), \ldots, x_n(t)\right), \\ \qquad\vdots \\ \frac{dx_n(t)}{dt} = f_n\left(t, x_1(t), x_2(t), \ldots, x_n(t)\right), \\ x_1(a) = s_1,\ x_2(a) = s_2,\ \ldots, x_n(a) = s_n. \end{cases}$$

The idea is to not be afraid of the notation, write everything as vectors, then *do exactly the same thing as for the one dimensional case!* That's right, there is nothing new but notation:

Let

$$\boldsymbol{X} = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix}, \quad \boldsymbol{X'} = \begin{bmatrix} x'_1 \\ x'_2 \\ \cdots \\ x'_n \end{bmatrix}, \quad \boldsymbol{F} = \begin{bmatrix} f_1 \\ f_2 \\ \cdots \\ f_n \end{bmatrix}, \quad \boldsymbol{S} = \begin{bmatrix} s_1 \\ s_2 \\ \cdots \\ s_n \end{bmatrix}.$$

Then we want to find $\boldsymbol{X}$ such that

$$\begin{cases} \boldsymbol{X'} = \boldsymbol{F}(t, \boldsymbol{X}) \\ \boldsymbol{X}(a) = \boldsymbol{S}. \end{cases}$$

Recasting Single ODE Methods

The framework is the same as for a single ODE: we know $\boldsymbol{X}(t)$, and want to find $\boldsymbol{X}(t+h)$. The step is defined by approximating $\boldsymbol{X}$ by its linearization. Quite simply we let

$$\boldsymbol{X}(t+h) \leftarrow \boldsymbol{X}(t) + h\boldsymbol{F}(t, \boldsymbol{X}(t)).$$

In fact, our general strategy of using Taylor's Theorem also carries through with out change. That is we can use the k^{th} order method to step as follows:

$$\boldsymbol{X}(t+h) \leftarrow \boldsymbol{X}(t) + h\boldsymbol{X'}(t) + \frac{h^2}{2}\boldsymbol{X''}(t) + \ldots + \frac{h^k}{k!}\boldsymbol{X^{(k)}}(t).$$

While we're at it, Runge Kutta works as well:

$$\begin{cases} \boldsymbol{K_1} = h\boldsymbol{F}(t, \boldsymbol{X}) \\ \boldsymbol{K_2} = h\boldsymbol{F}(t+h, \boldsymbol{X} + \boldsymbol{K_1}) \\ \boldsymbol{X}(t+h) \leftarrow \boldsymbol{X}(t) + \frac{1}{2}\boldsymbol{K_1} + \frac{1}{2}\boldsymbol{K_2}. \end{cases}$$

It's Only Systems

The fact we can simply solve systems of ODEs using old methods is good news. It allows us to solve higher order ODEs. For example, consider the following problem: given $f, a, c_0, c_1, \ldots, c_n$, find $x(t)$ such that

$$\begin{cases} x^{(n)}(t) = f\left(t, x(t), x'(t), \ldots, x^{(n-1)}(t)\right), \\ x(a) = c_0,\, x'(a) = c_1,\, x''(a) = c_2,\, \ldots,\, x^{(n-1)}(a) = c_{n-1}. \end{cases}$$

We can put this in terms of a system by letting

$$x_0 = x,\, x_1 = x',\, x_2 = x'',\, \ldots,\, x_{n-1} = x^{(n-1)}.$$

Then the ODE plus these $n-1$ equations can be written as

$$\begin{cases} x'_{n-1}(t) = f\left(t, x_0(t), x_1(t), x_2(t), \ldots, x_{n-1}(t)\right) \\ x'_0(t) = x_1(t) \\ x'_1(t) = x_2(t) \\ \vdots \\ x'_{n-2}(t) = x_{n-1}(t) \\ x_0(a) = c_0,\ x_1(a) = c_1,\ x_2(a) = c_2,\ \ldots,\ x_{n-1}(a) = c_{n-1}. \end{cases}$$

This is just a system of ODEs that we can solve like any other.

Note that this trick also works on systems of higher order ODEs. That is, we can transform any system of higher order ODEs into a (larger) system of first order ODEs.

Example 2. Rewrite the following system as a system of first order ODEs:

$$\begin{cases} x''(t) = (t/y(t)) + x'(t) - 1 \\ y'(t) = \frac{1}{(x'(t)+y(t))} \\ x(0) = 1,\ x'(0) = 0, y(0) = -1 \end{cases}$$

We let $x_0 = x$, $x_1 = x'$, $x_2 = y$, then we can rewrite as

$$\begin{cases} x'_1(t) = (t/x_2(t)) + x_1(t) - 1 \\ x'_0(t) = x_1(t) \\ x'_2(t) = \frac{1}{(x_1(t)+x_2(t))} \\ x_0(0) = 1,\ x_1(0) = 0, x_2(0) = -1 \end{cases}$$

It's Only Autonomous Systems

In fact, we can use a similar trick to get rid of the time-dependance of an ODE. Consider the first order system

$$\begin{cases} x'_1 = f_1\left(t, x_1, x_2, \ldots, x_n\right), \\ x'_2 = f_2\left(t, x_1, x_2, \ldots, x_n\right), \\ x'_3 = f_3\left(t, x_1, x_2, \ldots, x_n\right), \\ \qquad \vdots \\ x'_n = f_n\left(t, x_1, x_2, \ldots, x_n\right), \\ x_1(a) = s_1,\ x_2(a) = s_2,\ \ldots, x_n(a) = s_n. \end{cases}$$

We can get rid of the time dependance and thus make the system *autonomous* by making t another variable function to be found. We do this by letting $x_0 = t$, then adding the equations $x'_0 = 1$, and $x_0(a) = a$, to get the system:

$$\begin{cases} x_0' = 1, \\ x_1' = f_1\left(x_0, x_1, x_2, \ldots, x_n\right), \\ x_2' = f_2\left(x_0, x_1, x_2, \ldots, x_n\right), \\ x_3' = f_3\left(x_0, x_1, x_2, \ldots, x_n\right), \\ \qquad \vdots \\ x_n' = f_n\left(x_0, x_1, x_2, \ldots, x_n\right), \\ x_0(a) = a,\ x_1(a) = s_1,\ x_2(a) = s_2,\ \ldots, x_n(a) = s_n. \end{cases}$$

Autonomous systems are nice in that we can write them in the more elegant form:

$$\begin{cases} \boldsymbol{X'} = \boldsymbol{F}\left(\boldsymbol{X}\right) \\ \boldsymbol{X}\left(a\right) = \boldsymbol{S}. \end{cases}$$

But autonomous systems are really nice because we can talk about their *phase curves*.

Essentially, you can consider $\boldsymbol{X}$ to be the position of a particle in R^n. This particle moves over time. For an autonomous sytem of ODEs, the movement of the particle is only dependent on its current position, and not on the time. Thus one can conceivably draw the *phase curve*, which is the path of the particle, as a curve in R^n.

Stability

Maybe we will have time to consider stability:

Suppose you had an exact method of solving an ODE, but had the wrong data. That is, you were trying to solve

$$\begin{cases} \frac{dx(t)}{dt} = f\left(t, x(t)\right), \\ x(a) = c, \end{cases}$$

but instead fed the wrong data into the computer, which then solved exactly the ODE

$$\begin{cases} \frac{dx(t)}{dt} = f\left(t, x(t)\right), \\ x(a) = c + \epsilon. \end{cases}$$

This solution can diverge from the correct one because of the different starting conditions. We illustrate this with the usual example

$$\frac{dx(t)}{dt} = x.$$

The curves for different starting conditions diverge.

However, if we instead consider the ODE

$$\frac{dx(t)}{dt} = -x,$$

which has solution $x(t) = x(0)e^{-x}$, we see that differences in the initial conditions become immaterial.

It turns out there is a simple test for stability. If $f_x > \delta$ for some positive δ, then the ODE is unstable; if $f_x < -\delta$ for a positive δ, the equation is stable. Some ODEs fall through this test, however.

We can prove this without too much pain. Define $x(t,s)$ to be the solution to

$$\begin{cases} \frac{dx(t)}{dt} = f\left(t, x(t)\right), \\ x(a) = s, \end{cases}$$

for $t \geq a$.

Then unstability means that

$$\lim_{t\to\infty} \left| \frac{\partial}{\partial s} x(t,s) \right| = \infty.$$

Think about this in terms of the curves for our simple example $x' = x$.

If take the derivative of the ODE we get

$$\begin{aligned} \frac{\partial}{\partial t} x(t,s) &= f(t, x(t,s)), \\ \frac{\partial}{\partial s}\frac{\partial}{\partial t} x(t,s) &= \frac{\partial}{\partial s} f(t, x(t,s)), \quad i.e., \\ \frac{\partial}{\partial s}\frac{\partial}{\partial t} x(t,s) &= f_t(t, x(t))\frac{\partial t}{\partial s} + f_x(t, x(t,s))\frac{\partial x(t,s)}{\partial s}. \end{aligned}$$

By the independence of t, s the first part of the RHS is zero, so

$$\frac{\partial}{\partial s}\frac{\partial}{\partial t} x(t,s) = f_x(t, x(t,s))\frac{\partial x(t,s)}{\partial s}.$$

We use continuity of x to switch the orders of differentiation:

$$\frac{\partial}{\partial t}\frac{\partial}{\partial s} x(t,s) = f_x(t, x(t,s))\frac{\partial}{\partial s} x(t,s).$$

Defining $u(t) = \frac{\partial}{\partial s} x(t,s)$, and $q(t) = f_x(t, x(t,s))$, we have

$$\frac{\partial}{\partial t} u(t) = q(t)u(t).$$

The solution is $u(t) = ce^{Q(t)}$, where

$$Q(t) = \int_a^t q(r)\mathrm{d}r.$$

If $q(r) = f_x(r, x(r,s)) \geq \delta > 0$, then $\lim Q(t) = \infty$, and so $\lim u(t) = \infty$. But note that $u(t) = \frac{\partial}{\partial s} x(t,s)$, and thus we have unstability.

Similarly if $q(r) \leq -\delta < 0$, then then $\lim Q(t) = -\infty$, and so $\lim u(t) = 0$, giving stability.