# How Systems of First Order ODE's Arise

A system of first order ordinary differential equations is a family of $n$ ordinary differential equations in $n$ unknown functions $y_1(t),\ \cdots,\ y_n(t)$. To be ordinary (as opposed to partial) differential equations, only ordinary (as opposed to partial) derivatives of the unknown functions may appear. That is, the unknown functions are functions of a single variable, here called $t$. In order for each equation to be first order, the highest order derivative that appears in the equation must be one. So only $y_j(t),\ y_j'(t),\ 1 \le j \le n$ may appear in the equation and $y_j^{(m)}(t),\ 1 \le j \le n,\ m \ge 2$ may not. Some $y_j'(t),\ 1 \le j \le n$ must appear in each equation.

Systems of ODE's tend to arise by some combination of two basic mechanisms. First the original problem may involve the rates of change of more than one quantity. For example, in the predator-prey and competing species problems one studies how the population sizes of two (or more) species sharing some common ecosystem depend on time. In linear circuit problems one studies the behaviour of complex electrical circuits built from "linear circuit elements" like resistors and capacitors. The unknown are the currents $I_\ell$ in the various branches of the circuit and the charges $Q_j(t)$ on the various capacitors. The equations come from Kirchhoff's laws that state the total voltage around any closed loop in the circuit must be zero and that the total current entering any node of the circuit must be zero.

The second mechanism is the conversion of one higher order equation

$$y^{(n)}(t) = F\big(y(t), y'(t), \cdots, y^{(n-1)}(t)\big)$$

into a system of first order equations by the simple expedient of defining

$$y_j(t) = y^{(j)}(t), \qquad 0 \le j \le n-1$$

For each $0 \le j \le n-2$

$$\tfrac{d}{dt} y_j(t) = \tfrac{d}{dt}\tfrac{d^j}{dt^j} y(t) = \tfrac{d^{j+1}}{dt^{j+1}} y(t) = y_{j+1}(t)$$

and for $j = n-1$

$$\begin{aligned}\tfrac{d}{dt} y_{n-1}(t) &= \tfrac{d}{dt}\tfrac{d^{n-1}}{dt^{n-1}} y(t) = y^{(n)}(t) = F\big(y(t), y'(t), \cdots, y^{(n-1)}(t)\big)\\ &= F\big(y_0(t), y_1(t), \cdots, y_{n-1}(t)\big)\end{aligned}$$

So, the system

$$\begin{aligned}\tfrac{d}{dt}y_0(t) &= y_1(t)\\ \vdots \quad & \quad \vdots\\ \tfrac{d}{dt}y_{n-2}(t) &= y_{n-1}(t)\\ \tfrac{d}{dt}y_{n-1}(t) &= F\big(y_0(t), y_1(t), \cdots, y_{n-1}(t)\big)\end{aligned}$$

is equivalent to the original higher order system. The is, for each solution of the higher order equation, there is a corresponding solution of the linear system and vice versa.

This trick enables us to use any algorithm for generating, numerically, approximate solutions to systems of first order ode's to also generate approximate solutions to higher order ode's. And, typically, it is an easy matter to adapt an algorithm designed for handling one first order ode into an algorithm for handling systems of first order ode's. It suffices to put indices on the appropriate variables.

One common setting in which both mechanisms arise is that of particle mechanics. Supoose that we have a system of $N$ particles with $m_i$ and $\vec{r}_i(t)$ being the mass and position at time $t$, respectively, of particle number $i$. Then Newton's law

$$m_i\vec{r}_i^{\,\prime\prime} = \vec{F}_i(\vec{r}_1, \cdots, \vec{r}_N)$$

is a system of $3N$ second order ordinary differential equations in the $3N$ unknowns $\vec{r}_1, \cdots, \vec{r}_N$. To solve this system by computer one can convert them into the system

$$\begin{aligned}\vec{r}_i^{\,\prime} &= \vec{v}_i\\ \vec{v}_i^{\,\prime} &= \tfrac{1}{m_i}\vec{F}_i(\vec{r}_1, \cdots, \vec{r}_N)\end{aligned}$$

of $6N$ first order ode's in the $3N$ unknown positions $\vec{r}_1, \cdots, \vec{r}_N$ and $3N$ unknown velocities $\vec{v}_1, \cdots, \vec{v}_N$.