Consider the following nonlinear coupled system, where $x_1(t)$ and $x_2(t)$ evolve according to

$$
\dot{x}_1 = f_1(x_1, x_2), \quad \dot{x}_2 = f_2(x_1, x_2),
$$

for given functions $f_1$ and $f_2$, and also given initial conditions for $x_1$ and $x_2$. By defining the vector $\mathbf{x}(t) = (x_1(t), x_2(t))^T$ and the function $\mathbf{F} = (f_1, f_2)^T$, this system may be recast in the form $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x})$ with some initial condition $\mathbf{x}(0) = \mathbf{x}_0$. Note that this form is the nonlinear version of $\dot{\mathbf{x}} = A\mathbf{x}$, where $A$ is a matrix.

**Existence and uniqueness theorem (for $n$ dimensions)**

Consider the initial value problem $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}), \mathbf{x}(0) = \mathbf{x}_0$. Suppose that $\mathbf{F}$ is continuous and all its partial derivatives $\partial \mathbf{F}/\partial x_j$ (for $j = 1, \ldots, n$) are continuous in some open connected set $D \subseteq \mathbb{R}^n$. Then for $\mathbf{x}_0 \in D$, the initial value problem has solution $\mathbf{x}(t)$ on some time interval $t \in (-\tau, \tau)$ about $t = 0$, and the solution is unique.

Two important corollaries:

1. Two trajectories cannot intersect for a two-dimensional system, as otherwise the two trajectories would propagate from the same initial condition.

2. If a trajectory forms a closed curve $C$ then any other trajectory that is within $C$ remains within $C$ forever.

**Numerical solution**

The numerical methods for evolving scalar systems all extend to vector systems! For a time step $h > 0$, times $t_n = nh$ and corresponding numerical solution $\mathbf{x}_n \approx \mathbf{x}(t_n)$ (where $\mathbf{x}_0$ is the initial condition), the fourth-order Runge-Kutta method is

$$
\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),
$$

where between each time step, the vectors $k_j$ are defined as

$$
\begin{align*}
    k_1 &= h\mathbf{F}(\mathbf{x}_n), \\
    k_2 &= h\mathbf{F}(\mathbf{x}_n + k_1/2), \\
    k_3 &= h\mathbf{F}(\mathbf{x}_n + k_2/2), \\
    k_4 &= h\mathbf{F}(\mathbf{x}_n + k_3).
\end{align*}
$$

For Problem Set 4, you will need to write code for the fourth-order Runge-Kutta method in two-dimensions. The one-dimensional version of the code (in MATLAB) is uploaded on Stellar, which may be of use!