CDS 110a Wednesday 5 October 2005

Review of Linear Algebra and Ordinary Differential Equations (ODE's)

1. ***Last week:*** Matrices and vectors **`[with Matlab]`**
   a. Matrix-vector multiplication, matrix multiplication/addition, scalars
   b. Identity matrix, matrix inverses, singular matrices
   c. Eigenvalues, eigenvectors and diagonalization
   d. A brief note on Jordan forms
   e. Functions of matrices, matrix exponential
2. ***Today:*** Systems of first-order linear ODE's
   a. Scalar/decoupled case
   b. Matrix notation for coupled equations
   c. General solution by exponentiation
   d. $S + N$ decomposition and utility of the Jordan form
   e. Inhomogeneous equations with constant offsets
   f. Driven linear systems
   g. Integrating ODE's in **`Matlab`**
3. Examples (in recitations)
4. References:
   a. *Linear Algebra and its Applications*, G. Strang
   b. *Differential Equations and Dynamical Systems*, L. Perko
   c. **`http://www.mathworld.com`**

Systems of first-order ODE's
The simplest example an ODE is a first-order, scalar equation:

$$\dot{x} = ax.$$

Here let's assume $x \in R,\ a \in R$ (nothing really changes if they are complex). In your youth you have see the method of solution,

$$\frac{d}{dt}x = ax \rightarrow \frac{dx}{x} = a\,dt,$$

$$\int_{x_i}^{x_f} dx\,x^{-1} = a\int_{t_i}^{t_f} dt,$$

$$\ln\left(\frac{x_f}{x_i}\right) = a(t_f - t_i),$$

$$x_f = x_i \exp(a(t_f - t_i)).$$

Some simple things to notice at this point are that if $a < 0$ then x decreases to zero as $t_f \rightarrow +\infty$, and if $a > 0$ then x blows up exponentially (exercise: what if a is complex?).

As we mentioned in class on Monday, a higher-order ODE can always be converted into a system of coupled first-order ODE's by expanding the set of variables. For example, if we start with a single second-order equation in x,

$$m\ddot{x} + b\dot{x} + kx = 0,$$

we can convert by adding $\dot{x}$ as a distinct variable. Then,

$$\frac{d}{dt}\dot{x} = -\frac{b}{m}\dot{x} - \frac{k}{m}x,$$
$$\frac{d}{dt}x = \dot{x},$$

which we can write in "state space form" as

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \end{bmatrix}.$$

(Exercise: can we play the same trick if we start with $\dot{x} + x^2 = 0,$ by adding x^2 as a new variable?) One often writes,

$$\frac{d}{dt}\vec{x} = A\vec{x}, \quad \vec{x} \equiv \begin{bmatrix} x \\ \dot{x} \end{bmatrix},$$

where in Monday's terminology $\vec{x}$ is the “state vector” and A specifies the dynamics. Noting that the second-order equation we started with is essentially Newton's equation $F = ma$ for a damped mass-spring system, we see that systems of coupled equations will be ubiquitous in controls problems. Of course, even in problems without “inertia” (acceleration) terms, it is common to consider systems of coupled first-order ODE's from the outset. For example, if our state is the position of a point on the R^2 plane and our dynamics is a simple rigid rotation about the origin at frequency $\omega,$ we have

$$\frac{d}{dt}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}.$$

(Exercise: using methods from later in today's lecture, verify that this A matrix gives rise to rotation of the state vector.)

Note that in general if we have the dynamics written in state space form and A is a diagonal matrix, the ODE's aren't actually coupled to one another:

$$\frac{d}{dt}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_{xx} & 0 \\ 0 & a_{yy} \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}$$

is equivalent to

$$\dot{x} = a_{xx}x,$$
$$\dot{y} = a_{yy}y.$$

Obviously we can solve these equations independently to arrive at

$$x(t) = x(0)\exp(a_{xx}t),$$
$$y(t) = y(0)\exp(a_{yy}t).$$

This immediately leads us to one common strategy for integrating systems of first-order ODE's even when the equations are coupled (A has some non-zero off-diagonal matrix elements), by transforming to the eigenbasis of $A.$ Applying linear algebra concepts from last week, we can see that the state-space form of the dynamical equations

$$\frac{d}{dt}\vec{x} = A\vec{x}$$

is preserved under an invertible linear transformation P of the state space,

$$\vec{x} \mapsto \vec{x}' \equiv P^{-1}\vec{x}, \quad \vec{x} = P\vec{x}',$$

$$A \mapsto A' \equiv P^{-1}AP, \quad A = PA'P^{-1}.$$

Here's a simple proof:

$$\frac{d}{dt}(P\vec{x}') = PA'P^{-1}(P\vec{x}'),$$

$$P\frac{d}{dt}\vec{x}' = PA'\vec{x}',$$

$$\frac{d}{dt}\vec{x}' = A'\vec{x}'.$$

(Note that it is crucial to assume that P^{-1} exists.) Hence if A happens to be diagonalizable, we can choose P to be the matrix of eigenvectors such that

$$A' = P^{-1}AP = D,$$

where D is the diagonal matrix of eigenvalues. As a result, we have

$$\frac{d}{dt}\vec{x}' = D\vec{x}'$$

as a set of decoupled equations that we can solve independently. After doing so, we can obtain our solution in the original coordinates by computing

$$\vec{x}(t) = P\vec{x}'(t),$$

which is of course quite simple to do. It follows that for any diagonalizable $A,$ the solutions $\vec{x}(t)$ take the form of linear combinations of exponentials of $t.$ Note that if A has complex eigenvalues, however, these may combine into trig functions. (Exercise: use this strategy to solve

$$\frac{d}{dt}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix},$$

and comment on why the solutions $x(t)$ and $y(t)$ are always real (assuming $x(0)$ and $y(0)$ are real) even though the eigenvalues of A are complex.)

What happens if A is not diagonalizable? For any A whatsoever, we can always make use of the general closed-form solution

$$\vec{x}(t) = \exp(At)\vec{x}(0),$$

where of course this is to be understood as a matrix exponential. Note how this nicely captures the procedure we followed above, in the case of diagonalizable A:

$$
\begin{aligned}
\vec{x}(t) &= \exp(At)\vec{x}(0) \\
&= \exp(PDP^{-1}t)\vec{x}(0) \\
&= \left[1 + PDP^{-1}t + \frac{1}{2}(PDP^{-1}t)^2 + \frac{1}{3!}(PDP^{-1}t)^3 + \cdots\right]\vec{x}(0) \\
&= \left[1 + PDP^{-1}t + P\frac{1}{2}(Dt)^2P^{-1} + P\frac{1}{3!}(Dt)^3P^{-1} + \cdots\right]\vec{x}(0) \\
&= P\exp(Dt)P^{-1}\vec{x}(0).
\end{aligned}
$$

(Make sure you can follow the step from line 3 to line 4.) We can read our eigenbasis procedure from the last line - transform the initial conditions into the eigenbasis, multiply each term by the exponential of an eigenvalue times t, and then transform back to the original basis. If A is not diagonalizable, we can still (in principle) use this closed-form solution if we can find some other way to compute the matrix exponential.

Exponentiation of non-diagonalizable matrices

Consider the case $F = ma$ with $F = 0$ (a free particle). Then

$$\ddot{x} = 0,$$

which in state space form reads

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \end{bmatrix}.$$

Then our closed-form solution reads,

$$\vec{x}(t) = \exp(At)\vec{x}(0), \quad At = \begin{bmatrix} 0 & t \\ 0 & 0 \end{bmatrix}.$$

Note that this is a nilpotent matrix! Explicitly,

$$At = \begin{bmatrix} 0 & t \\ 0 & 0 \end{bmatrix}, \quad (At)^2 = \begin{bmatrix} 0 & t \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & t \\ 0 & 0 \end{bmatrix} = 0.$$

Hence,

$$
\begin{aligned}
\exp(At) &= 1 + At + \frac{1}{2}(At)^2 + \cdots \\
&\to 1 + At = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}.
\end{aligned}
$$

As a result $\vec{x}(t) = (1 + At)\vec{x}(t)$, which reads componentwise

$$
\begin{aligned}
x(t) &= x(0) + \dot{x}(0)t, \\
\dot{x}(t) &= \dot{x}(0).
\end{aligned}
$$

This agrees with what we would write down based on Ph1. Right? Note that the A matrix in this case is not diagonalizable and that the closed-form solution for $\vec{x}(t)$ does not have the form of a linear combination of exponentials of t.

Things will not be so simple in general. However, here's a trick that can often be used to simplify calculation of the matrix exponential. Suppose A is not diagonalizable, but that we can see a way to split it into a symmetric part and a nilpotent part:

$$A = S + N, \quad S = S^T, \quad N^r = 0.$$

For example,

$$A = \begin{bmatrix} 2 & 0 \\ -1 & 2 \end{bmatrix} = S + N = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix},$$

$$N^2 = \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix}\begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix} = 0.$$

If furthermore we can verify that S and N commute,

$$[S, N] = SN - NS = 0,$$

(as they do in our simple example) then computation of the matrix exponential is simplified as follows:

$$\begin{aligned} \exp(At) &= \exp(St + Nt) \\ &= \exp(St)\exp(Nt) \\ &= P\exp(Dt)P^{-1}\left(\sum_{j=1}^{r-1} \frac{1}{j!}(Nt)^r\right). \end{aligned}$$

Note that the first step (breaking up the exponential) relies on the assumption that S and N commute. Also note that the symmetry of S guarantees that we can find an invertible P to diagonalize it. In principle, we can always transform to a basis in which A' splits up into a diagonal matrix and nilpotent matrix - the Jordan canonical form:

$$\begin{aligned} A &= TJT^{-1} = T(D + N)T^{-1}, \\ \exp(At) &= T\exp(Dt + Nt)T^{-1} \\ &= T\exp(Dt)\exp(Nt)T^{-1}. \end{aligned}$$

Note that the detailed structure of the Jordan form guarantees that D and N here commute. But finding the invertible linear transformation that does this is not always so easy. (Exercise: above we saw that diagonalizable A give rise to integrated solutions $\vec{x}(t)$ that are sums of exponentials, whereas our free particle $F = ma$ example gave us a linear dependence on t ; what is the most general form of dependence on t for the integrated solutions of $\frac{d}{dt}\vec{x} = A\vec{x}$?)

Inhomogeneous and driven systems

Sometimes one encounters inhomogenous linear ODE's:

$$\dot{x} = ax + b,$$

where $\{a, b\} \in R.$ A simple way to solve this scalar case is to change to a new variable,

$$x' \equiv x + \frac{b}{a},$$
$$\dot{x}' = \dot{x}.$$

Then

$$\dot{x}' = ax + b = a\left(x' - \frac{b}{a}\right) + b$$
$$= ax',$$

which can be solved by simple exponentiation. Similarly, we may encounter an inhomogeneous system of coupled ODE's

$$\frac{d}{dt}\vec{x} = A\vec{x} + \vec{b},$$

with $\left\{\vec{x}, \vec{b}\right\} \in R^n$. In order to play the analogous, trick we need to find a solution $\vec{c}$ to the equation

$$A\vec{c} = \vec{b}.$$

If we can do that, then

$$\vec{x}' \equiv \vec{x} + \vec{c},$$
$$\frac{d}{dt}\vec{x}' = \frac{d}{dt}\vec{x} = A\vec{x} + \vec{b} = A(\vec{x}' - \vec{c}) + \vec{b} = A\vec{x}'.$$

This transformation strategy is often referred to as 'translating the equilibrium to the origin,' since we note that the state $\vec{x} = -\vec{c}$ is a stationary solution of the dynamics:

$$\frac{d}{dt}\vec{x} \mapsto A(-\vec{c}) + \vec{b} = 0.$$

If A is not invertible, however, a different procedure must be used.

For example, we might consider $F = ma$ with constant $F = f$. In state space form,

$$\frac{d}{dt}\vec{x} = A\vec{x} + \vec{b}, \quad A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} 0 \\ f/m \end{bmatrix}.$$

Looking at the second row,

$$\frac{d}{dt}\dot{x} = f/m,$$

we can easily integrate this to find

$$\dot{x}(t) = \dot{x}(0) + \frac{f}{m}t.$$

We can then substitute this into the first row to find the scalar 'driven' ODE,

$$\frac{d}{dt}x = \dot{x} = \dot{x}(0) + \frac{f}{m}t.$$

We can in turn integrate this to find

$$\int_{x(0)}^{x(t)} dx = \int_0^t dt\left(\dot{x}(0) + \frac{f}{m}t\right),$$
$$x(t) = x(0) + \dot{x}(0)t + \frac{f}{2m}t^2.$$

It is straightforward to convince yourself that a solution like this cannot correspond to

$$\frac{d}{dt}\vec{x}' = A\vec{x}'$$

with $\vec{x}' \in R^2 = \vec{x} + \vec{c}$ for constant $\vec{c}$ (do this as an exercise; think about what you can and cannot do with nilpotent 2×2 matrices).

This brings us finally to a general consideration of linear systems with additive time-dependent driving terms:

$$\frac{d}{dt}\vec{x} = A\vec{x} + \vec{b}(t),$$

where $\vec{x} \in R^n$ and $\vec{b}(t) \in R^n$ is an arbitrary time-dependent driving term. A general solution for the initial value problem can be written as

$$\vec{x}(t) = \exp(At)\vec{x}(0) + \exp(At)\int_0^t ds\, \exp(-As)\vec{b}(s)$$

$$= \exp(At)\vec{x}(0) + \int_0^t ds\, \exp(A(t-s))\vec{b}(s).$$

We can can some intuition regarding the form of this solution by thinking about the superposition principle for linear dynamics. Note that if

$$\vec{x}(0) = \vec{x}_a(0) + \vec{x}_b(0),$$

then

$$\vec{x}(t) = \exp(At)\vec{x}(0) = \exp(At)\vec{x}_a(0) + \exp(At)\vec{x}_b(0).$$

Obviously this works for any number of additive terms in the initial condition. Likewise, for any given value of the state vector at time t, we can split $\vec{x}(t)$ up into any number of parts and think of each one as having ‘come from’ some corresponding component of the initial condition:

$$\vec{x}(t) = \sum_j \vec{x}_j(t) = \sum_j \exp(At)\vec{x}_j(0),$$

where

$$\vec{x}_j(0) \equiv \exp(-At)\vec{x}_j(t).$$

Note that this works even if A itself is not invertible. Hence, rewriting the first form of the general solution above as

$$\vec{x}(t) = \exp(At)\left\{\vec{x}(0) + \int_0^t ds\, \exp(-As)\vec{b}(s)\right\},$$

we can interpret the integral as signifying that the effects of the driving term are incorporated by adding terms $\sim \exp(-As)\vec{b}(s)ds$ to the ‘effective’ initial condition for the state vector. Each of these terms looks like the vector obtained by evolving $\vec{b}(s)\,ds$ backwards in time back to 0.

In previous lectures we have seen the input-output state-space form for linear dynamics,

$$\frac{d}{dt}\vec{x} = A\vec{x} + B\vec{u},$$

$$\vec{y} = C\vec{x}.$$

Here $\vec{u}(t)$ is a completely arbitrary function of time, and we can identify $B\vec{u}(t)$ with the driving term $\vec{b}(t)$ considered above.

Integrating ODE's in Matlab

The general state-space form of dynamics is

$$\frac{d}{dt}\vec{x} = f(x,t),$$

where $f(x,t)$ may be a nonlinear function. In Matlab, one can numerically solve the initial value problem for such a system using 'ODE solvers' such as **`ode45`**. In order to do this, you first need to write a Matlab function that evaluates $f(x,t)$ and pass a handle to that function file as an argument to **`ode45`** (see **`help ode45`** from the Matlab prompt). Note that there are other Matlab ODE solver such as **`ode15`**s, which are tailored to handle stiff differential equations (involving a very wide range of timescales). (Exercise: give this a try for $\dot{x} = \log\left(\sqrt{x+1}\right)$ for $t_i = 0,\ t_f = 10$ and $x(0) = 0$).

Of course, for linear systems you can always use the closed-form solutions described above and the built-in matrix exponentiation routine **`expm`**.