**Coupled ODE’s**

**System of Linear Equations**

Suppose we have equations:



We can write this in matrix form, where x = (x1, x2)T, and:



The solution to this equation is, formally:



Explicitly, we’d introduce the eigenvalue decomposition of the matrix A:



Note this implies:



(see Appendix) so u1, u2 are the eigenvectors of the matrix A, and λ1, λ2 are the eigenvalues. Then we could write our formal solution as:



But otherwise, going back to the ODE, we’d say:



then define = U†x, and we can say:



and then multiplying both sides by U, we get:



Since U = (1|2|…|3), we can recognize (0) = (**x**(0)∙1, **x**(0)∙2) as the projection of the initial (time t = 0) vector onto the first eigenvector and second eigenvector respectively. Also, we can write **x**(t) as (see Appendix)



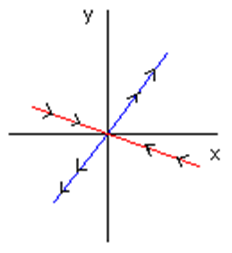
So there we are:



So 1 and 2 are the eigenvectors of A, and λ1 and λ2 are the eigenvalues. And we can read this as the projection of the initial conditions on the first eigenvector times the development of that eigenvector + the projection of the initial conditions on the second eigenvector times the development of that eigenvector. And on an individual coordinate basis, we can write:



Anyway, note whichever eigenvalue is positive governs the long time limit. We can represent the positive eigenvalue eigenvector with the blue line and the negative one with the red line.



Finally, note that a stable fixed point would be had if all the arrows pointed towards the origin (both eigenvalues are negative). And unstable fixed point would be had if they all pointed away (both are positive). And if neither, then don’t know what that’s called. So if the initial conditions started on the blue line, then the solution would proceed along towards infinity on the blue line. And if initial conditions start on the red line, then solution would be driven towards the origin.

A quick note on the Wronskian. We can check the linear independence of our two solutions as follows:



**System of Linear Equations Reprised in Dirac notation**

Suppose we have equations:



We can write this in matrix form, where x = (x1, x2)T, and:



Now consider these QM-esque manipulations…we’ll introduce the eigenbasis of A, denoted |en>. Then,



where we define the projection of |x> onto |en> as n = <en|x>. We can solve this to get:



So then,



So our solution is:



Back in matrix form, this says that our solution is:



Or even more directly,



**A question**

Going back to:



we will want to solve:



Can we do this?



So for there to be non-zero solutions to this equation, we’d need:



which is precisely the typical eigenvalue equation. So that’s cool.

**Example 1**

Let’s look at this guy:



We can solve these by diagonalizing performing,



Define



Eigenvalues are:



Eigenvectors are:



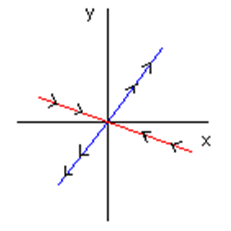
So,



So then our solution is:



We can represent this in a sort of phase diagram. **e**+ is the blue line and **e**- the red line.



More information can be gleaned if we look at



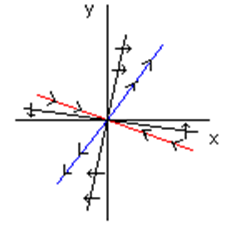
We see that along the line,



any solution has a horizontal slope. And along the line,



any solution has a purely vertical slope. So now our graph looks like this. We don’t seem to add much information this way though.



**Example 2**

We’ll find the general solution to:



General solution is of the form:



where λ1, λ2 are the eigenvalues of M, and **e**1, **e** 2 are eigenvectors of M (going to write as **e**, instead of , so as to distinguish between unnormalized eigenvectors, **e**, and normalized eigenvectors ). The eigenvalues of M are determined by the equation:



So eigenvalues are λ1 = -1 and λ2 = 2. Now we need to find the eigenvectors corresponding to each eigenvalue. So we solve:



Multiplying this matrix equation out gives us the following two equations:



Now these two equations are linearly dependent, so we may keep one and drop the other one. Doesn’t matter which one we keep and which one we drop. I’ll drop the first one and keep the second.



Let ey = 1, say, and then we’d have:



and our general eigenvector could be written as:



Filling in λ1 and λ2 explicitly, our two eigenvectors are:



So then, our general solution is:



This is our answer. But we could write it a little prettier, by redefining A as 2A (A is an arbitrary constant so it doesn’t matter). Then we’d have:



**Example 3**

We’ll find the general solution to:



General solution is of the form:



where λ1, λ2 are the eigenvalues of M, and **e**1, **e**2 are the eigenvectors of M. The eigenvalues of M are determined by the equation:



So eigenvalues are:



Now we need to find the eigenvectors corresponding to each eigenvalue. So we solve:



Multiplying this matrix equation out gives us the following two equations:



Now these two equations are linearly dependent, so we may keep one and drop the other one. Doesn’t matter which one we keep and which one we drop. I’ll drop the first one and keep the second.



Let ey = 1, say, and then we’d have:



and our general eigenvector could be written as:



Filling in λ1 and λ2 explicitly, our two eigenvectors are:



So then, our general solution is:



We could stop here. Or we could change A 🡪 4A and B 🡪 4B without affecting the generality of our expression because A and B are arbitrary constants. If so, then we’d have:



Only benefit of this is that it looks a little nicer. Might want to separate into real and imaginary parts:



Let’s write A = C + Di and B = C – Di (i being the imaginary unit). Then,



Finally, this can be written as:



**Example 4**

We could do a Laplace transform. Say we have:



A Laplace transform gives,



Now need the inverse of this matrix,



Inverse Laplace transform is:



So one at a time,



and



**Adding Diagonal Terms**

Let’s consider equation of this form. As a side note, if we have an equation of form:



then we can write this as:



Can we get rid of the c term? Let’s define: y = e-ctx. Then,



**Time-Dependent A(t)**

Let’s go back to:



If we do a series expansion, we’d have:



What if A is something like:



Then this would suggest,



Does this satisfy the equation?



But what if A is time-dependent? Then series expansion would be:



So, this works out to:



If A(t) commutes with itself at all times, then we could take the time-ordering off.

**Coupled Linear inhomogeneous any order ODE`s**

Let’s move on to more general coupled equations. Consider first second order coupled ODE's, with constant coefficients:



Where L is an operator of the form L = Σn(anDn), where D is the derivative operator. Apply L3 to the top equation and L1 to the bottom one:



Since the operators are linear with constant coefficients, they commute, and you can subtract the two equations to obtain:



Which can be written



Going back and doing the analogous procedure for y1 we'd find:



We can perhaps write these more suggestively as:



which evoke Cramer's rule from linear algebraic equations. We just have to interpret these equations correctly. You can perhaps also reduce a general linear system of equations to a linear second order equation and maybe use power series with a coupled recursion relation? You can also use the laplace transform for initial condition systems.

**Appendix**

Let’s review some matrix multiplication identities. Let:



And let V be some matrix of appropriate dimensionality. Then we can say:



And some matrix eigenvalue properties. Say M is given by:



Then its eigenvalues are:



where we let:



We can remember that the sign in the root function must be positive because a real symmetric matrix can’t have imaginary eigenvalues.

Note a derivative issue. Can’t use chain rule on derivatives. Consider A2.



I guess A and Aʹ have to commute. So this means:

