Math 110 - Fall 05 - Lectures notes # 28 - Nov 2 (Wednesday)

Now we begin Chapter 5, on eigenvalues, eigenvectors,
and diagonalizability. You might want to review Appendices
D and E on complex numbers and polynomials.
We will have several motivating examples for eigenvalues:
  Vibrations:
     How do we know if this building will fall down in an earthquake?
     How does my cell-phone "tune in" to the signal?
  Data analysis:
     How does Google decide the order in which to show pages?

We'll start with vibrations, using a simple example of a mass
hanging from a spring, and work up to a whole building
(which we'll do in less detail!). Suppose the mass is hanging
at rest. Push it up, hold it steady for a moment, and let go.
ASK&WAIT: What happens?
Let's write down an equation for the motion of the spring.
Our starting point is F = ma from freshman physics.
Let m   = mass of the spring
    x(t) = distance of mass from resting position (positive means up)
           at time t.
For the force from the spring when it is compressed or stretched,
we use F = -k*x, where k>0 is called the "spring constant".
This just means that the spring pushes down (pulls up) if the
mass moves up (down), and does so with a force proportional
to how far the mass has moved. So F = ma turns into
   -k*x(t) = m*d^2 x(t)/dt^2
or
   d^2 x(t)/dt^2 = -k/m * x(t)
Since we expect a periodic motion up and down, let's just try
one of the simplest periodic functions we know, x(t) = cos(w*t),
and plug it in to see if it works:
   -w^2 * cos(w*t) = -k/m * cos(w*t)
which is satisfied if w = sqrt(k/m). w is called the
"frequency of vibration" or "resonant frequency".
At t=0, x(t) = 1 and d x(t)/dt = w*sin(w*t) = 0,
so x(t) = cost(w*t) satifies the initial conditions
(mass starts with 0 velocity, pushed up distance 1 from
the resting position)

Now suppose we take the top of the spring and move it up and down

at some frequency f; this is our simplest model of an earthquake.
What happens to the mass? The equation changes to

$$d^2 x(t)/d^2t = -k/m * x(t) + \sin(f*t)$$

where sin(f*t) is the periodic motion of the earthquake.
Suppose we start at rest (at t=0, x(t) = 0 and dx(t)/dt = 0);
What is the solution? There are two cases, depending on whether
f=w (assume both f and w are positive, for simplicity)

If f neq w, we can confirm that the solution is the sum of
two sine waves, at frequencies f and w:

$$x(t) = c1 * \sin(w*t) + c2 * \sin(f*t)$$
$$\text{where } c1 = -(f/w)/(w^2-f^2) \text{ and } c2 = 1/(w^2-f^2)$$

(Homework: plug in and confirm!)
Since the sines are bounded by 1 in absolute value,
|x(t)| is bounded by |c1| + |c2| = 1/|w*(w-f)| for all t.
So the motion x(t) is bounded for all t,
but something clearly happens as f get closer to w:
the bound on |x(t)| gets larger and larger.

If f = w, we can confirm that the solution is

$$x(t) = c * (1 - w*t) * \sin(w*t)$$
$$\text{where } c = 1/(2*w^2)$$

(Homework: plug in and confirm!)
Now, x(t) is not bounded as t grows: the size of the vibrations
grow proportionally to t.
ASK&WAIT: What does this mean, physically?

Physically, we say the mass-spring system is in resonance.
It would clearly not be comfortable to be riding on top of
the mass when this happened, which should make you think
of what might happen in an earthquake.

Now for some linear algebra: Instead of a single mass and spring,
lets try hanging 3 vertically (picture), mass 3 at the bottom,
mass 2 in the middle, and mass 1 at the top, with three springs
(wth spring 1 at the top, spring 2 in the middle, spring 3 at
the bottom). For simplicity each mass will be equal, but spring i
will have spring constant k_i, and x_i will be distance of mass i

from its resting position (up is positive).
Now we just write down F=ma three times, once for each mass:
```
   m*d^2 x_1(t)/t^2 =
      -k_1 * x_1
         ... force from spring 1, which is compressed a distance x_1
      -k_2 * (x_1 - x_2)
         ... force from spring 2, which is stretched by x_1 - x_2
   m*d^2 x_2(t)/t^2 =
      -k_2 * (x_2 - x_1)
         ... force from spring 2, which is compressed by x_2 - x_1
      -k_3 * (x_2 - x_3)
         ... force from spring 3, which is stretched by x_2 - x_3
   m*d^2 x_3(t)/t^2 =
      -k3 * (x_3 - x_2)
         ... force from spring 3, which is compressed by x_3 - x_2
```
We write these 3 equations as a single matrix equation, by letting

```
  x(t) = [x_1(t)]  d^2 x(t)/dt^2 = [d^2 x_1(t)/dt^2]
         [x_2(t)]                  [d^2 x_2(t)/dt^2]
         [x_3(t)]                  [d^2 x_3(t)/dt^2]

  K = -1/m * [ k_1 + k_2    -k_2        0    ]
             [   -k_2     k_2 + k+3   -k_3   ]
             [     0        -k_3       k_3   ]
```

and so
```
          d^2 x(t)/dt^2 = K * x(t).
```

As above, we try to find a solution of the form x(t) = cos(w*t) * y,
where now y is some nonzero constant vector and w is some scalar; we need
to determine both w and y. Plug in to get

```
   d^2 x(t)/dt^2 = -w^2*cos(w*t)*y = K*x(t) = cos(w*t)*K*y
```

The only way this can be true for all t is for

```
    -w^2*y = K*y
```

Def 1: Let A be an n by n matrix with entries from field F.
Suppose there is a scalar lambda and a nonzero vector y, all
with entries from F, such that K*y = lambda*y.
Then lambda is called an eigenvalue and y is called an

```
eigenvector of K. We sometimes also say (lambda,y) is
an eigenpair, or "y is lambda's eigenvector" or
"lambda is y's eigenvalue".

Def 2: Let T: V -> V be a linear operator over a field F.
Then if there is a scalar lambda in F and a nonzero
vector y in V such that T(y) = lambda*y, lambda is called
an eigenvalue and y is an eigenvector. (This definition
applies even when V is infinite dimensional.)

Note: Any nonzero multiple of an eigenvector is also an eigenvector.

Recall: det(A) = 0 iff A is noninvertible iff A has a nonzero nullspace

Thm 1: lambda is an eigenvalue of A if and only if det(A - lambda*I) = 0.
Proof: A*y = lambda*y iff (A-lambda*I)*y = 0, i.e. iff
       A-lambda*I has nonzero nullspace.

Ex: 2x2 case: A = [a_11 a_12]  each entry from a field F
                  [a_21 a_22]
    0 = det(A-lambda*I) = det([a_11 - lambda  a_12         ])
                              [a_21           a_22 - lambda])
              = (a_11 - lambda)*(a_22 - lambda) - a_12*a_21
              = lambda^2 - lambda*(a_11 + a_22) + (a_11*a_22 - a_12*a_21)
    a quadratic equation in lambda. What are its roots?

    Suppose F = reals and A = [1 1 ; 1 1], then
      det(A - lambda*I) = lambda^2 -2*lambda; there are two real roots, 0 and 2
ASK&WAIT: What are the corresponding eigenvectors?
          Are they linearly independent?

    Suppose F = reals and A = [1 1 ; 0 1], then
      det(A - lambda*I) = lambda^2 -2*lambda + 1 = (lambda - 1)^2;
          there is one real root, 1, with "multiplicity 2"
      What are the eigenvectors? They must satisfy A*y = y, or
          y_1 + y_2 = y_1 and y_2 = y_2; The first equation implies
          y_2 = 0, y_1 can be any nonzero scalar. In other words all
          eigenvectors are of the form [y_1 ; 0], i.e. multiples of [1;0]
          Thus there is only 1 independent eigenvector

    Suppose F = reals and A = [0 1 ; -1 0], then
      det(A - lambda*I) = lambda^2 + 1; there are no real roots,
```

```
          but complex ones (with complex eigenvectors)

    Suppose F = rationals and A = [0 1 ; 2 0], then
      det(A - lambda*I) = lambda^2 - 2; there are no rational roots,
          but real ones (with real eigenvectors)

Returning to our vibration example, we note that all numbers
must be real for this to make physical sense. The solution was
    x(t) = cos(w*t) * y  where K*y = -w^2*y. Thus -w^2 is an
    eigenvalue and y is an eigenvector of K.  In other words
    we need that the eigenvalue -w^2 is nonpositive, in order
    for w to be real. This is indeed the case for this K.
    Suppose now that we "shake" the masses according to the
    following equation, much as we shook the single mass:
       d^2 x(t)/dt^2 = K*x + sin(f*t)*y
    Let's try to find a solution of the form x(t) = z(t)*y,
    where z(t) is a scalar function:
       d^2 x(t)/dt^2 = d^2 z(t)/dt^2 * y
                     = K*z(t)*y + sin(f*t)*y
                     = z(t)*(-w^2)*y + sin(f*t)*y
                     = (-w^2*z(t) + sin(f*t)) * y
    For this to be true, we must have
      d^2 z(t)/dt^2 = -w^2 * z(t) + sin(f*t)
    which is exactly the same scalar equation we had before,
    with bounded solutions when w neq f, and "resonance"
    (unbounded solutions) when w = f, i.e. when the
    eigenvalue -w^2 = -f^2.

Now what about deciding if a building will fall down in an
earthquake? This is a complicated question, but here is one
approach. The structural parts of a building, such as the steel
beams, may be modeled as masses and springs, of course connected
in three dimensions. Writing down F = ma for all the pieces
of the building (a process called "finite element analysis"
in civil and mechanical engineering) again leads to an
equation
    d^2 x(t)/dt^2 = K * x(t)
where K is typically enormous, with its dimension n usually
many thousands or even sometimes millions. Again we get
solutions of the form x(t) = cos(w*t)*y, where -w^2 is
an eigenvalue of K and y is an eigenvector. Now K will have n
eigenpairs (lambda_1,y_1), (lambda_2,y_2),..., (lambda_n,y_n),
```

corresponding to the roots of its characteristic polynomial of degree n. If the building is shaken, say by an earthquake, at a frequency f that matches one of these eigenvalues (lambda_i ~ -f^2 for some i), then the building will undergo large vibrations, and be in danger of falling down.
So what an engineer does is look at seismographic recordings of known earthquakes, and see what the usual frequencies of earthquakes are. (They are typically between 1 and 10 vibrations per second)
If any of these match your buildings eigenvalues, you're in trouble. Hopefully the design engineer will discover this while designing the building and change the design before the building is built.

There are several famous bridges that have started vibrating because of this resonance. The Tacoma Narrows bridge collapsed in 1940 because of resonance caused by wind rather than earthquakes. In the on-line video at

www.enm.bris.ac.uk/research/nonlinear/tacoma/tacoma.html

the sine-wave oscillations are clearly evident.

The Millennium Bridge for pedestrians, which opened over the River Thames in London in 2000, was closed shortly thereafter because the pedestrian traffic set off a resonance that required a redesign and retrofit. The web site

www.arup.com/MillenniumBridge

includes a video of the vibrations. An analysis of this phenonmenon was recently published by Steve Strogatz of Cornell in Nature (3 Nov 2005 issue, www.nature.com).

We are currently engaged in a project to put motion sensors onto the Golden Gate Bridge to detect onset of such vibrations caused by wind (which will not cause the GGB to collapse, but to possibly move several 10s of feet east to west, causing serious traffic problems).

Resonance is not always bad, as in these examples. Indeed, all radio, TV and other telecommunications depends on it, because it is how tiny electrical signals are detected and amplified. For example, in your cell phone the tiny electrical signal in the antenna causes a quartz crystal to resonate mechanically, which is how the signal is detected. There is so little energy, and so much damping ("friction" in the material) that the vibrations remain microscopic, and indeed cell-phone designers would like to make them larger, because this would save (battery) power, not just in cell-phones but in all telecommunication devices. A number of research groups are engaged in trying to design new resonators that resonate as strongly as possible at cell-phone frequencies (gigaHertz), which amounts to adjusting their shape to make their eigenvalues have certain target values.

We will consider the Google example after we have more background.

Now we go back to the formal treatment of eigenvalues and eigenvectors.

Thm 2: det(A - lambda*I) is a polynomial in lambda of degree n. (This polynomial is called the characteristic polynomial of A).

Thus the roots of the characteristic polynomial (those which are in the field F of entries of A, the only scalars we are formally allowed to talk about) are the eigenvalues.

Ex: Suppose A is diagonal. Then A - lambda*I is also diagonal and det(A-lambda*I) = product_{i=1 to n} (A_ii - lambda), which is a polynomial with roots A_11,...,A_nn, so the eigenvalues are just the diagonal entries.

ASK&WAIT: What if A is upper triangular?

Proof 1: One of our equivalent definitions of a determinant, the "explicit formula", was to multiply out the determinant explicitly, getting a big polynomial in the matrix entries, where we multiply n entries together in each term of the polynomial. Since each entry of A - lambda*I is either a_ij or a_ii - lambda, multiplying n such factors together gives us terms of degree at most n. Multiplying all the diagonal entries a_ii - lambda gives us a term (-1)^n * lambda^n, the only one of degree n in lambda.

Proof 2: We can expand along the first row and use the induction hypothesis that all the n-1 by n-1 determinants are polynomials of degree at most n-1. (Homework!)

The next goal is to show that any polynomial can be factored into a product of factors of the form (x-r_i) using its roots r_i.

Some basic facts about polynomials are reviewed in Appendix E.

Lemma 1: If r is a root of polynomial p(x), i.e. p(r) = 0,
then we can write p(x) = (x-r)*q(x), where q(x) is a
polynomial of one lower degree than p(x).

Proof: Just divide p(x) by (x-r) getting
    p(x)=(x-r)*q(x) + remainder
and note that the remainder is a constant that must be 0.
See appendix E.

Def 3: A polynomial p(x) has a root r of multiplicity k if we
can write p(x) = (x-r)^k * q(x) where q(x) is another polynomial,
and k is the largest integer for which this is true.

Lemma 2: If p(x) has degree n and q(x) has degree m then
p(x)*q(x) has degree n+m

Proof: Exercise!

Thm 3: An n by n matrix can have at most n eigenvalues,
counting multiplicities. In other words, if r_1 through
r_k are distinct roots of p(x), where r_i has
multiplicity m_i, then m_1 + m_2 + ... + m_k <= n.

Proof of Thm 3: Let p(x) = det(A - lambda*I) be the characteristic
polynomial. By assumption (x-r_1)^m_1 through (x-r_k)^m_k
all divide p(x). We claim that we can therefore factor
p(x) = (x-r_1)^m_1 * (x-r_2)^m_2 * ... * (x-r_k)^m_k * s(x),
where s(x) is another polynomial. Assuming for a moment
that this is true, then by Lemma 2 the degree of p is the sum
of the degrees of its factors, and so
degree(p) = n = m_1 + m_2 + ... + m_k + degree(s) >= m_1 + ... + m_k
as desired.

To prove that p(x) factors as described, we can use induction.
on the m_i. We start writing p(x) = (x-r_1)^m_1 * q(x) which we
are given. Since (x-r_2)^m_2 also divides p(x), we get p(r_2)=0,
and so q(r_2)=0 (since (r_2 - r_1) neq 0).
So by Lemma 1, q(x) = (x-r_2)*q2(x).
Thus p(x) = (x-r_1)^m_1*(x-r_2)*q2(x). Since (x-r_2)^m_2 divides p(x),
we get (x-r_2)^(m_2-1) divides (x-r_1)^m_1*q2(x), so q2(r2)=0 and

we can factor out another (x-r_2). We keep repeating this for
each factor of the form (x-r_i).

Corollary 1: The number of distinct eigenvalues of an n by n
matrix A is at most n.

Proof: If r_1,...,r_k are distinct roots, Thm 3 implies
n >= m_1 + ... + m_k >= k.

As pointed out above, not all polynomials have n roots, counting multiplicities, if we insist the roots have to be in the same field as the polynomial coefficients. For example, polynomials with rational entries may have irrational or complex roots.

Def 4: A polynomial with coefficients over F "splits" if all its roots are in F.

Ex: x^2 - 2 splits over R but not Q. x^2 + 1 splits over C but not R.

Def 5: A field F with the property that all polynomials with coefficients from F have all roots in F is called algebraically closed.

The field of complex numbers is algebraically closed, as shown in Appendix D. In Ma114 you may study other algebraically closed fields. In some of our results we will assume F is algebraically closed, in others we will not. We will always assume that our fields are infinite (i.e. no finite fields like Z_2).

We briefly consider eigenvalues of linear operators T:V->V for infinite dimension T. One can imagine that one no longer necessarily gets a finite set of eigenvalues. Here is an example:

```
Let V = {"smooth" functions from R -> R,
          i.e. they have derivatives of all orders}
Let T(f) = f', i.e. taking the derivative. Then
T(f) = lambda*f means f'(x) = lambda*f(x), a
differential equation with solution f(x) = c*exp(lambda*x).
Thus every real number is an eigenvalue of T.
```

Recall that many results of Chapter 3 (and even Chapter 4) came down to understanding and using a single "matrix decomposition", namely the LU decomposition. There is a similar matrix decomposition for

Chapter 5, which we now discuss. Unlike the LU decomposition, it does not exist for every matrix (for reasons including the fact that all roots of a polynomial may not exist in the field F). But it exists for "most" matrices over C, and is very useful. We will call it an eigendecomposition (the book also says "diagonalization").

Def 6: An eigendecomposition of an n by n matrix A is the expression A = Q * Lambda * Q^{-1} where Q is an invertible matrix, and Lambda = diag(lambda_1,...,lambda_n) is diagonal. If this decomposition exists, we say A is diagonalizable.

Recall the definition of two matrices being similar: A = Q*B*Q^{-1} means A and B are similar. So another way to define diagonalizability is that A is diagonalizable if it is similar to a diagonal matrix.

Thm 4: An eigendecomposition A = Q * Lambda * Q^{-1} exists if and only if

```
  (1) for all i, lambda_i is an eigenvalue of A
  (2) column i of Q (call it q_i) is the eigenvector of A
      belonging to lambda_i
  (3) The n eigenvectors {q_1,...,q_n} are linearly independent
```

Proof: Assume the eigendecomposition exists.
Multiply A = Q*Lambda*Q^{-1} on the right by Q to get the equivalent equality A*Q = Q * Lambda. Multiply this by the j-th standard basis vector e_j to get

```
  column j of A*Q = (A*Q)*e_j = (Q*Lambda)*e_j = column j of Q*Lambda
        (A*Q)*e_j = A*(Q*e_j)   ... by associativity
                  = A*q_j       ... where q_j = column j of Q
                  = (Q * Lambda)*e_j
                  = Q * (Lambda*e_j) ... by associativity
                  = Q * (lambda_j*e_j) ... since Lambda is diagonal
                  = lambda_j * (Q * e_j)  ... factoring out lambda_j
                  = lambda_j * q_j
```

or A*q_j = lambda_j * q_j, i.e. for all j, lambda_j is an eigenvalue and q_j is its eigenvector. The fact that Q = [q_1,q_2,...,q_n] is assumed to be invertible means that its columns are linearly independent (otherwise Q would have a nonzero nullspace).

Conversely, if (1) and (2) hold, the above argument shows that A*q_j = lambda_j * q_j can be written as A*Q = Q*Lambda.

```
Now (3) implies that Q is invertible, and so we can multiply
A*Q = Q*Lambda on the right by Q^{-1} to get
A*Q*Q^{-1} = A = Q*Lambda*Q^{-1}, the desired eigendecomposition.

It is easy to compute the characteristic polynomial of a diagonalizable
matrix:
  det(A - x*I) = det(Q*Lambda*Q^{-1} - x*I)
               = det(Q*Lambda*Q^{-1} - x*Q*I*Q^{-1})
                   ... since Q*I*Q^{-1}=I
               = det(Q*(Lambda - x*I)*Q^{-1})
                   ... factoring out Q and Q^{-1}
               = det(Q) * det(Lambda - x*I) * det(Q^{-1})
                   ... since det(A*B) = det(A)*det(B)
               = det(Q) * det(Q^{-1}) * det(Lambda - x*I)
               = det(Lambda - x*I)
                   ... since det(Q) * det(Q^{-1}) = 1
               = product_{i=1 to n} (lambda_i - x)
                   ... since Lambda - x*I is diagonal
whose roots are evidently lambda_1 through lambda_n, as expected.

Now we go back to our first motivating problem, solving a system
of linear differential equations:
   d x(t)/dt = A * x(t)
where x(t) is a vector of unknown functions, A is an n by n matrix,
and f(t) is a vector of known functions. Suppose we also know x(0).
If A is diagonalizable, this is very easy to solve: Write
   d x(t)/dt = Q*Lambda*Q^{-1} * x(t)
and multiply through by Q^{-1} to get
   Q^{-1} * d x(t)/dt = Lambda * Q^{-1} * x(t)
Define a new vector of unknowns y(t) = Q^{-1) * x(t). Then
y(0) = Q^{-1} * x(0) and
   d y(t)/dt = Lambda * y(t)
Since Lambda is diagonal we can write this as n separate equations:
   d y_1(t)/dt = lambda_1 * y_1(t)
   d y_2(t)/dt = lambda_2 * y_2(t)
              ...
   d y_n(t)/dt = lambda_n * y_n(t)
each of which is easy to solve: y_i(t) = y_i(0) * exp(lambda_i * t)
Given each component of y(t), we can therefore get the solution
of the original equation
 x(t) = Q * y(t)
```

```
= Q * [ y_1(0) * exp(lambda_1 * t) ]
                 ...
      [ y_n(0) * exp(lambda_n * t) ]

= Q * [ exp(lambda_1*t)       0               0           ] * [y_1(0)]
      [        0       exp(lambda_2*t)        0           ]   [y_2(0)]
      [                      ...                          ]   [ ...  ]
      [        0             ...        exp(lambda_n*t)   ]   [y_n(0)]

= Q * [ exp(lambda_1*t)       0               0           ] * Q^{-1}*x(0)
      [        0       exp(lambda_2*t)        0           ]
      [                      ...                          ]
      [        0             ...        exp(lambda_n*t)   ]
```

To solve the more general system d x(t)/dt = A*x + f(t), where
f(t) is a vector function, the same idea also reduces it to solving
n scalar differential equations, which we know how to do from Ma53:

```
   d x(t)/dt = A * x(t) + f(t)
             = Q*Lambda*Q^{-1} * x(t) + f(t)
```

Multiply through by Q^{-1} to get

```
   Q^{-1} * d x(t)/dt = Lambda * Q^{-1} * x(t) + Q^{-1}*f(t)
```

Define a new vector of unknowns y(t) = Q^{-1) * x(t) and
known vector g(t) = Q^{-1}*f(t), so y(0) = Q^{-1} * x(0) and

```
   d y(t)/dt = Lambda * y(t) + g(t)
```

Since Lambda is diagonal we can write this as n separate equations:

```
   d y_1(t)/dt = lambda_1 * y_1(t) + g_1(t)
              ...
   d y_n(t)/dt = lambda_n * y_n(t) + g_n(t)
```

each of which we know how to solve from Ma53.
Given each component of y(t), we can therefore get the solution
of the original equation via x(t) = Q * y(t).

We have already seen that some matrices are not diagonalizable.
For example, we saw that A = [1 1 ; 0 1] had only one eigenvector,
whereas we would need two independent eigenvectors to diagonalize it.
Eventually (Chapter 7) we will see how to generalize eigendecompositions
to this case (using the Schur decomposition or Jordan Canonical Form),
but in the meantime let us state some simple theorems about when a
matrix is diagonalizable. (We will have more such theorems later.)

Thm 5: If lambda_1,...,lambda_k are distinct eigenvalues of A,
then their eigenvectors q_1,...,q_k are linearly independent.

Corollary 2: If all the eigenvalues of A are distinct, then A
is diagonalizable.

Proof of Corollary 2: If the n eigenvectors q_1,...,q_n
are independent, then the matrix Q = [q_1,...,q_n] is
nonsingular, and so by Thm 4, A is diagonalizable.

Proof of Thm 5: We use proof by contradiction.
Suppose q_1,...,q_k are dependent, that is

```
    sum_{i=1 to k} a_i*q_i = 0
```

for some a_i, not all zero. Assume without loss of generality
that a_k neq 0. Now multiply the above equation by A - lambda_1*I:

```
   0 = (A-lambda_1*I) * sum_{i=1 to k} a_i*q_i
     = sum_{i=1 to k} a_i* (A - lambda_1*I)*q_i
         ... since A-lambda_1*I is a linear transformation
     = sum_{i=1 to k} a_i* (lambda_i - lambda_1)*q_i
         ... since A*q_i = lambda_i*q_i
     = sum_{i=2 to k} a_i* (lambda_i - lambda_1)*q_i
         ... since the i=1 term is zero
     = sum_{i=2 to k} a'_i* q_i
         ... where a'_i = a_i*(lambda_i-lambda_1)
```

where each a'_i is nonzero iff a_i is nonzero, since
(lambda_i - lambda_1) is nonzero for i>1.
Now take the last equation and multiply it by A - lambda_2*I.
The same logic shows

```
  0 = sum_{i=3 to k} a''_i * q_i
```

where each a''_i = (lambda_i-lambda_2)*(lambda_i-lambda_1)*a_i
is nonzero iff a_i is nonzero. Continue multiplying by

```
  A - lambda_3*I, A - lambda_4*I,..., A - lambda_{k-1}*I,
```

until we get

```
  0 = (lambda_k-lambda_1)*...*(lambda_k-lambda_{k-1})*a_k*q_k
```

All the factors (lambda_k - lambda_i) are nonzero because
the all the lambda_i are distinct. Thus a_k=0, but this contradicts
the assumption that a_k neq 0.

A matrix can be diagonalizable even when the eigenvalues are not distinct.
ASK&WAIT: Can you give a simple example? Where all the eigenvalues are
identical?
Here is a more general theorem about diagonalizability that takes
multiple eigenvalues into account.

```
Def 7: Let A be a matrix with eigenvalue lambda.
Then E_lambda = Null(A - lambda*I) = {x: A*x = lambda*x}
is called the eigenspace of T corresponding to lambda.

Note that if x in E_lambda, then A*x = lambda*x.
ASK&WAIT: Does this means E_lambda = {all eigenvectors of A for lambda}?

Ex: If A = I, with eigenvalue 1, then E_1 = Null(I - 1*I) = F^n,
all vectors.

Ex: If A = [2 0 0] then E_2 = span{[1],[0]} and E_3 = span{[0]}
           [0 2 0]                [0],[1]                [0]
           [0 0 3]                [0],[0]                [1]

Thm 6: If lambda is an eigenvalue of A with multiplicity m,
then dim(E_lambda) <= m. In other words, there are at most
m linearly independent eigenvectors for eigenvalue lambda.

Proof: Let {v_1,...,v_k} be an ordered basis for E_lambda.
We want to show k <= m.  Extend {v_1,...,v_k} to an ordered basis
of F^n: {v_1,...,v_n}.  Let Q = [v_1,...,v_n],
which must be a nonsingular matrix.
Then the j-th column of A*Q is
   A*Q*e_j = A*(Q*e_j) = A*v_j = lambda*v_j if j <= k.
In other words, A*Q = [v_1,...,v_k,x_{k+1},...,x_n]
where x_{k+1},...,x_n are some other vectors that we
don't need to know more about. Then when j <= k,
the j-th column of Q^{-1}*A*Q is
  (Q^{-1}*A*Q)*e_j = Q^{-1}*(A*Q*e_j)
                   = Q^{-1}*(lambda*v_j)
                   = lambda * Q^{-1}*(v_j)
                   = lambda * Q^{-1}*(Q*e_j)
                   = lambda * (Q^{-1}*Q)*e_j
                   = lambda * (I)*e_j
                   = lambda * e_j
In other words
                      k    n-k
   Q^{-1}*A*Q = [ lambda*I  B ] k
                [     0     C ] n-k

where B and C are matrices we don't need to know more about.
Thus the characteristic polynomial p(x) of A is
```

```
  det(A - x*I) = det(Q^{-1})*det(A-x*I)*det(Q)
               = det(Q^{-1}*(A-x*I)*Q)
               = det([ (lambda-x)*I   B  ])
                     [      0       C-x*I])
               = det((lambda-x)*I)*det(C-x*I)
               = (lambda-x)^k*det(C-x*I)
so (lambda-x)^k is a factor of p(x). Thus, by
the definition of multiplicity, k <= m, as desired.

Thm 7. A is diagonalizable if and only if
   n = sum_{i=1 to k} dim(E_{lambda_i})
where lambda_1,...,lambda_k are all the
distinct eigenvalues of A.

Proof: First, suppose that A is diagonalizable:
   A = Q*Lambda*Q^{-1}
where we have numbered the eigenvalues and
eigenvectors (columns of Q) so that
the first n_1 eigenvalues are equal with common value lambda_1,
the next  n_2 eigenvalues are equal with common value lambda_2 neq lambda_1,
and so on. Thus the first  n_1 columns of Q, call them
  q^(1)_1,...,q^(1)_{n_1}
are n_1 independent eigenvectors for lambda_1,
the next n_2 columns of Q, call them
  q^(2)_1,...,q^(2)_{n_2}
are n_2 independent eigenvectors for lambda_2, and so on. Thus
  E_{lambda_i} = Null(A - lambda_i*I) = span(q^(i)_1 ,..., q^(i)}_{n_i})
and
  dim(E_{lambda_i}) = n_i.
and
  sum_{i=1 to k} dim(E_{lambda_i}) = sum_{i=1 to k} n_i = n
as desired.

Second, suppose n = sum_{i=1 to k} dim(E_{lambda_i})
Define n_i = dim(E_{lambda_i}).  Let
   Q_i = [q^(i)_1 , ... , q^(i)_{n_i}]
be an n by n_i matrix whose columns are an ordered basis for E_{lambda_i}.
Let
   Q = [Q_1,...,Q_k]
be an n by n matrix. Then by construction
   A*Q = Q*Lambda
where Lambda is a diagonal matrix whose first n_1 entries are
```

```
lambda_1, next n_2 entries of lambda_2, and so on.
To prove A is diagonalizable, By Thm 4 all we still need to show is
that Q is invertible, i.e. its columns are independent,
i.e if some linear combination
  0 = sum_{i=1 to k}  sum_{j=1 to n_i} a_ij* q^(i)_j
then we need to show all the a_ij = 0. Note that each inner sum
      z_i = sum_{j=1 to n_i} a_ij * q^(i)_j
gives a vector z_i inside the subspace E_{lambda_i}, i.e. an eigenvector
for lambda_i. Thus
  0 = sum_{i=1 to k} z_i
But by Thm 5, this can only be true if each z_i = 0, since (nonzero)
eigenvectors corresponding to distinct eigenvalues are independent.
But by the independence of q^(i)_1,...,q^(i)_{n_i} in the sum for z_i,
all the coefficients a_ij must be zero.

We use diagonalizability to talk about computing functions of matrices.

If A is a matrix, A^i means A*A*...*A, (i factors of A).

Def 8: If p(x) = sum_{i=0 to d} p_i * x^i is a polynomial, then
for any square matrix
  p(A) = sum_{i=0 to d} p_i * A^i.

Thm 8: Suppose A is diagonalizable: A = Q * Lambda * Q^{-1}
with Lambda = diag(lambda_1,...,lambda_n).
Then p(A) = Q * p(Lambda) * Q^{-1}
          = Q * diag( p(lambda_1),...,p(lambda_n) ) * Q^{-1}
is also diagonalizable, with eigenvalues p(lambda_i)
and the same eigenvectors as A.

Proof:
    A^i = (Q*Lambda*Q^{-1})^i
        = (Q*Lambda*Q^{-1})*(Q*Lambda*Q^{-1})*...(Q*Lambda*Q^{-1})
               ... i copies of (Q*Lambda*Q^{-1})
        = Q*Lambda*(Q^{-1}*Q)*Lambda*(Q^{-1}*Q)*...*(Q^{-1}*Q)*Lambda*Q^{-1})
               ... associativity
        = Q*Lambda*(I)*Lambda*(I)*...*(I)*Lambda*Q^{-1})
               ... "cancelling" each Q^{-1}*Q
        = Q*Lambda^i*Q^{-1}
so
   p(A) = sum_{i=0 to d} p_i * A^i
        = sum_{i=0 to d} p_i * Q*Lambda^i*Q^{-1}
```

```
        = Q * (sum_{i=0 to d} p_i * Lambda^i ) * Q^{-1}
        = Q * p(Lambda) * Q^{-1}
Furthermore
   p(Lambda) = sum_{i=0 to d} p_i * Lambda^i
             = sum_{i=0 to d} p_i * diag(lambda_1^i , ... , lambda_n^i )
             = diag( sum_{i=0 to d} p_i * lambda_1^i , ... ,
                     sum_{i=0 to d} p_i * lambda_n^i )
             = diag( p(lambda_1) , ... , p(lambda_n) )

Corollary 3: (Cayley-Hamilton Thm for diagonalizable matrices)
Let p(x) be the characteristic polynomial of A = Q * Lambda * Q^{-1}.
Then p(A) = 0, the zero matrix.

Proof: Thm 8 implies
  p(A) = Q * p(Lambda) * Q^{-1}
       = Q * diag(p(lambda_1),...,p(lambda_n)) * Q^{-1}
       = Q * diag(0,...,0) * Q^{-1}
             ... since the eigenvalues are roots of p(x)
       = 0
(Eventually we will prove p(A)=0 for all A, even when A is not
diagonalizable, which is the complete statement of the
Cayley-Hamilton Thm.)

Now we consider functions defined by Taylor series, not just polynomials.
We will say that a sequence of matrices A^(k) converges to a limit matrix A
as k -> infinity, if each component A^(k)_ij converges to A_ij.
Suppose f(x) = sum_{i=0 to infinity} f_i * x^i is a Taylor series expansion
that converges in some region; this means that the limit of the following
sequence of polynomials converges:
    lim_{n -> infinity} sum_{i=0 to n} f_i * x^i
We will say that f(A) converges to a limit if the sequence of polynomials
    lim_{n -> infinity} sum_{i=0 to n} f_i * A^i
converges to a matrix.

Thm 9: Suppose f(x) = sum_{i=0 to infinity} f_i * x^i converges for
x equal to any of the eigenvalues of the diagonalizable matrix
A = Q*Lambda*Q^{-1}, where Lambda = diag(lambda_1,...,lambda_n).
Then f(A) converges, and is diagonalizable:
    f(A) = Q * diag( f(lambda_1,...,f(lambda_n) ) * Q^{-1}
with eigenvalues f(lambda_i) and the same eigenvectors as A.

Proof: Let f_N(x) = sum_{i=0 to N} f_i * x^i. Then
```

```
from Thm 8,
   f_N(A) = Q * diag(f_N(lambda_1),...,f_N(lambda_n)) * Q^{-1}
As N -> infinity, by assumption f_N(lambda_i) -> f(lambda_i).
Thus
   f_N(A) -> Q * diag(f(lambda_1),....f(lambda_n) * Q^{-1}
as desired.

Example: suppose d x(t)/dt = A*x(t), x(0) given.
with A = Q * diag(lambda_1,...,lambda_n) * Q^{-1}
diagonalizable. Recall that the solution is
    x(t) = Q * diag(exp(lambda_1*t),...,exp(lambda_n)*t) * Q^{-1} * x(0)
We can now rewrite this as
    x(t) = exp(t*A) * x(0)
by applying Thm 9 to the diagonalizable matrix
    t*A = Q * diag(t*lambda_1,...,t*lambda_n) * Q^{-1}
```