Math 110 - Fall 05 - Lectures notes # 11 - Sep 23 (Friday)

The next goal is to make explicit the connection between
matrices, familiar from Math 54, and linear transformations
T: V -> W between finite dimensional vectors spaces.
They are not quite the same, because the matrix that represents
T depends on the bases you choose to span V and W, and
the order of these bases:

Def: Let V be a finite dimensional vector space. An
ordered basis of V is a basis for V with an order:
{v_1,...,v_n}, where n=dim(V).

Ex: Let e_i = i-th standard basis vector (1 in i-th entry, 0 elsewhere)
Then the bases {e_1,e_2,e_3} and {e_2,e_3,e_1} are the same
(order in a set does not matter), but the ordered bases
{e_1,e_2,e_3} and {e_2,e_3,e_1} are different.

Def: For V = F^n, {e_1,...,e_n} is the standard ordered basis.
For P_n(F), {1,x,x^2,...,x_n} is the standard ordered basis.

Given ordered bases for V and W, we can express both vectors
in V and W, and linear transformations T:V->W as vectors and
matrices with respect to these ordered bases:

Def: Let beta = {v_1,...,v_n} be an ordered basis for V.
For any x in V, let x = sum_{i=1 to n} a_i*v_i be the unique
linear combination representing x. The coordinate vector
of x relative to beta, denoted [x]_beta, is

```
   [x]_beta = [ a_1 ]
              [ a_2 ]
              [ ... ]
              [ a_n ]
```

ASK & WAIT: What is [v_i]_beta ?
ASK & WAIT: Let V = P_5(F), beta = {1,x,x^2,x^3,x^4,x^5},
 and v = 3-6x+x^3. What is [v]_beta?
ASK & WAIT: If beta = {x^5, x^4, x^3, x^2, x, 1}?

Lemma: The mapping Beta: V -> F^n that maps x to [x]_beta is linear.
Proof: if x = sum_i a_i*v_i and y = sum_i b_i*v_i, then
         [x]_beta = [a_1;...;a_n], [y]_beta=[b_1;...;b_n] and

```
[x+y]_beta = [sum_i (a_i+b_i)*v_i]_beta   ... by def of x+y
           = [a_1+b_1 ; ... ; a_n+b_n ]   ... by def of []_beta
           = [a_1 ; ... ; a_n] + [b_1; ... ; b_n]
           = [x]_beta + [y]_beta          ... by def of []_beta
 Similarly, [c*x]_beta = c*[x]_beta
```

We need this representation of vectors in V as
coordinate vectors of scalars in order to apply T: V -> W
as multiplication by a matrix. We will also need to
represent vectors in W the same way.

Let beta = {v_1,...,v_n} and gamma = {w_1,...,w_m}
be ordered bases of V and W, resp. Let T: V -> W be linear.
Then there are unique scalars a_{ij} such that

```
    T(v_j) = sum_{i=1 to m} a_{ij}*w_i
```

These scalars will be the entries of the matrix representing T:

Def: Let T: V -> W be linear, V and W finite dimensional.
Using the above notation, the m x n matrix A with
entries a_{ij}, is the matrix representation of T in the ordered
bases beta and gamma. We write A = [T]_beta^gamma.
If V = W and beta = gamma, we write simply A = [T]_beta

Note that column j of A is [a_{1j};...;a_{mj] = [T(v_j)]_gamma

To see why we call A the matrix representation of T, let us use
it to compute y = T(x).
Suppose x = sum_{j=1 to n} x_j*v_j, so [x]_beta = [x_1;...;x_n]
is the coordinate vector for x. We claim the coordinate
vector for y is just gotten by multiplying by A:

```
  [y]_gamma = A * [x]_beta
```

To confirm this we compute:

```
     y = T(x) = T(sum_{j=1 to n} x_j*v_j)  ... by def of x
              = sum_{j=1 to n} x_j*T(v_j)  ... since T is linear
              = sum_{j=1 to n} x_j*(sum_{i=1 to m} a_{ij}*w_i)
                                           ... by def of T(v_j)
              = sum_{j=1 to n} sum_{i=1 to m} a_{ij}*x_j*w_i
                                           ... move x_j into sum
              = sum_{i=1 to m} sum_{j=1 to n} a_{ij}*x_j*w_i
                                           ... reverse order of sums
              = sum_{i=1 to m} w_i * (sum_{j=1 to n} a_{ij}*x_j)
                                           ... pull w_i out of inner sum
```

```
so
[y]_gamma = [ sum_{j=1 to n} a_{1j}*x_j ] = A * [ x_1 ] = A*[x]_beta
            [ sum_{j=1 to n} a_{2j}*x_j ]       [ x_2 ]   as desired
            [             ...           ]       [ ... ]
            [ sum_{j=1 to n} a_{mj}*x_j ]       [ x_n ]

Ex: T:R^2 -> R^4, T((x,y) = (x-y, 3*x+2*y, -2*x, 7*y)
    beta = standard basis for R^2, gamma = standard basis for R^4,
    so  T((1,0)) = (1;3;-2;0) and T((0,1)) = (-1;2;0;7), so
    A = [ 1 -1 ] (for brevity in these notes, we will sometimes use
        [ 3  2 ]  "Matlab notation": T = [ 1 -1 ; 3 2 ; -2 0 ; 0 7 ] )
        [-2  0 ]
        [ 0  7 ]

ASK & WAIT: What if beta = {e2,e1} and gamma = {e3 e4 e1 e2}?

Ex (continued): Suppose x = 3*e1 - e2; what is T(x)?
   what is [T(x)]_gamma, using standard bases?
   T(x) = T(3,-1) = (4,7,-6,-7)
   [T(x)]_gamma = A * [3;-1] = [4;7;-6;-7]

Ex: T: P_3(R) -> P_2(R), T(f(x)) = f'(x),
beta = {1, 1+x, x^2, x^3 }, gamma = {2 , x , x^2}
Then T(1) = 0, T(1+x) = 1 = (1/2)*2; T(x^2) = 2*x ; T(x^3) = 3*x^2
So T = [ 0 1/2 0 0 ]
       [ 0  0  2 0 ]
       [ 0  0  0 3 ]
ASK & WAIT: What is T if beta = { 1, x, x^2, x^3 }? If gamma={1,x,x^2}?

Having identified matrices with linear transformations between
two finite dimensional spaces with ordered bases, and recalling
that mxn matrices form a vector space, we will not be surprised
that all the linear transformations between any two vector
spaces is also a vector space:

Def: Let T and U be linear transformations from V -> W.
Then we define the new function T+U: V -> W by (T+U)(v) =T(v)+U(v)
and the new function c*T: V -> W by (c*T)(v) = c*T(v)

Thm: Using this notation, we have that
  (1) For all scalars c, c*T+U is a linear transformation
  (2) The set of all linear transformation from V -> W,
```

```
      is itself a vector space, using
      the above definitions of addition and multiplication by scalars
Proof:
  (1) (c*T+U)(sum_i a_i*v_i)
           = (c*T)(sum_i a_i*v_i) + U(sum_i a_i*v_i) ... by def of c*T+U
           = c*(T(sum_i a_i*v_i)) + U(sum_i a_i*v_i) ... by def of c*T
           = c*(sum_i a_i*T(v_i)) + sum_i a_i*U(v_i) ... since T,U linear
           = sum_i a_i*c*T(v_i) + sum_i a_i*U(v_i)
           = sum_i a_i*(c*T(v_i)+U(v_i))
           = sum_i a_i*(c*T+U)(v_i)                  ... by def of c*T+U
  (2) We let T_0, defined by T_0(v) = 0_W for all v, be the
      "zero vector" in L(V,W). It is easy to see that all the
      axioms of a vector space are satisfied. (homework!)

Def: L(V,W) is the vector space of all linear transformations
     from V -> W. If V=W, we write L(V) for short.

Given ordered bases for finite dimensional V and W, we get a
matrix [T]_beta^gamma for every T in L(V,W). It is natural
to expect that the operations of adding vectors in L(V,W)
(adding linear transformations) should be the same as adding
their matrices, and that multiplying a vector in L(V,W) by a
scalar should be the same as multiplying its matrix by a scalar:

Thm: Let V and W be finite dimensional vectors spaces with
ordered bases beta  and gamma, resp. Let T and U be in L(V,W).
Then
   (1) [T+U]_beta^gamma =   [T]_beta^gamma + [U]_beta^gamma
   (2) [c*T]_beta^gamma = c*[T]_beta^gamma
In other words, the function []_beta^gamma: L(V,W) -> M_{m x n}(F)
is a linear transformation.

Proof: (1) We compute column j of matrices on both sides
           and comfirm they are the same. Let beta = {v_1,...,v_n}
           and gamma = {w_1,...,w_m}. Then
        (T+U)(v_j) = T(v_j) + U(v_j) so
       [(T+U)(v_j)]_gamma = [T(v_j)]_gamma + [U(v_j)]_gamma
        by the above Lemma that shows the mapping x -> [x]_gamma
        was linear.
         (2) Similarly (c*T)(v_j) = c*T(v_j) so
             [(c*T)(v_j)]_gamma = [c*T(v_j)]_gamma
                                 = c*[T(v_j)]_gamma by the Lemma
```

```
            so the j-th columns of both matrices are the same

To summarize, what we have so far is this:
Given two ordered bases beta for V and gamma for W, we have
a one-to-one correspondence between
all linear transformations in L(V,W) and all matrices in M_{m x n}(F).
Furthermore, this 1-to-1 correspondence "preserves" operations
in each set:
  adding linear transformations <=> adding matrices
        [ T + U <=> [T]_beta^gamma + [U]_beta^gamma ]
  multiplying linear transformations by scalars
          <=> multiplying matrices by scalars
        [  c*T  <=> c*[T]_beta^gamma ]
  applying linear transformation to a vector
          <=> multiplying matrix times a vector
        [ y = T(x)  <=>  [y]_gamma = [T]_beta^gamma * [x]_beta  ]

There are several other important properties of and operations
on linear transformations, and it is natural to ask what they
mean for matrices as well (we write A = [T]_beta^gamma for short)
  null space N(T)    <=> set of vectors x such that A*x = 0
  range space R(T)   <=> all vectors of the form A*x
                     <=> all linear combinations of columns of A
  T being one-to-one <=> Ax = 0 only if x=0
  T being onto       <=> span(columns of A) = F^m
  T being invertible <=> A square and Ax=0 only if x=0

What about composition? If T: V -> W and U: W -> Z, then
consider UT: V -> Z, defined by UT(v) = U(T(v))

Thm: If T is in L(V,W) and U is in L(W,Z), then UT is in L(V,Z),
i.e. UT is linear.
Proof: UT( sum_i a_i*v_i ) = U(T( sum_i a_i*v_i ))  ... by def of UT
                           = U( sum_i a_i*T(v_i))   ... since T is linear
                           = sum_i a_i*U(T(v_i))    ... since U is linear
                           = sum_i a_i*UT(v_i)      ... by def of UT

Thm:
  (1) If T in L(V,W) and U1 and U2 are in L(W,Z), then
      T(U1+U2) = TU1 + TU2  (distributibity of addition over composition)
  (2) If T in L(V,W), U in L(W,Z), and A in L(Z,Q), then
      T(UA) = (TU)A is in L(V,Q)   (associativity of composition)
```

```
  (3) If T in L(V,W), I_V the identity in L(V,V) and I_W the identity in L(W,W),
      then T I_V = I_W T = T   (composition with identity)
  (4) If T in L(V,W) and U in L(W,Z), and c a scalar, then
      c(TU) = (cT)U = T(cU)  (commutativity and associativity of
      composition with scalar multiplication)
Proof: homework!

Now that we understand the properties of composition of linear transformations,
we can ask what operation on matrices it corresponds to:

Suppose T: V -> W and U: W -> Z are linear so UT: V -> Z
Let beta  = {v_1,...,v_n} be an ordered basis for V
    gamma = {w_1,...,w_m} be an ordered basis for W
    delta = {z_1,...,z_p} be an ordered basis for Z
with A = [U]_gamma^delta the p x m matrix for U,
     B = [T]_beta^gamma the m x n matrix for T,
     C = [UT]_beta^delta the p x n matrix for UT
Our question is what is the relationship of C to A and B?
We compute C as follows:
   column j of C = [UT(v_j)]_delta        ... by def of C
                 = [U(T(v_j))]_delta      ... by def of UT
                 = [U(sum_{k=1 to m} B_kj*w_k)]_delta  ... by def of T(v_j)
                 = [sum_{k=1 to m} B_kj*U(w_k)]_delta  ... by linearity of U
                 = [sum_{k=1 to m} B_kj* sum_{i=1 to p} A_ik * z_i]_delta
                         ... by def of U(w_k)
                 = [sum_{k=1 to m} sum_{i=1 to p} A_ik * B_kj * z_i]_delta
                         ... move B_kj inside summation
                 = [sum_{i=1 to p} sum_{k=1 to m} A_ik * B_kj * z_i]_delta
                         ... reverse order of summation
                 = [sum_{i=1 to p} z_i * (sum_{k=1 to m} A_ik * B_kj)]_delta
                         ... pull z_i out of inner summation
                 = [sum_{k=1 to m} A_1k * B_kj]   ... by def of []_delta
                   [sum_{k=1 to m} A_2k * B_kj]
                   [            ...           ]
                   [sum_{k=1 to m} A_mk * B_kj]
said another way, C_ij = sum_{k=1 to m} A_ik * B_kj

Def: Given the p x m matrix A, and the m x n matrix B we define their
matrix-matrix product (or just product for short) to be the p x n matrix C
with C_ij = sum_{k=1 to m} A_ik * B_kj

Thm: Using the above notation,
```

```
        [UT]_beta^delta = [U]_gamma^delta * [T]_beta^gamma

In other words, another correspondence between linear transformations
and matrices is:
    composition of linear transformations <=> matrix-matrix multiplication
```