Chapter 3: Linear Transformations (Operators) on Vector Spaces:

Think of linear transformations, or *operators*, on vector spaces as similar to functions. Basically, it is a *rule* that associates a vector in one space to a vector in another (or maybe the same) space.

DEFINITION: A linear transformation $A: X \to Y$ **from vector space *X* into vector space *Y* is linear if**

$$A(\alpha_1 x_1 + \alpha_2 x_2) = \alpha_1 A x_1 + \alpha_2 A x_2$$

for any vectors $x_1, x_2 \in X$.

DEFINITION: The **range space** of *A*, denoted $\mathcal{R}(A)$, is the set of all vectors $y_i \in Y$ such that for every $y_i \in \mathcal{R}(A)$ there exists an $x \in X$ such that $A(x) = y_i$.

If $\mathcal{R}(A)=Y$, the transformation (or mapping) is *onto*.
(i.e., if the range of A is the entire space)

If *A* maps elements of *X* to unique values in *Y*; that is, if

$$x_1 \neq x_2 \Rightarrow A(x_1) \neq A(x_2)$$

then *A* is *one-to-one*.

If *A* is one-to-one and onto, then it is invertible; i.e., A^{-1} **exists such that**

$$A^{-1}(A(x)) = x \qquad \text{(or } A^{-1}A = I\text{)}$$

<u>DEFINITION</u>: The null space of *A*, denoted $\mathcal{N}(A)$ **is the set of all vectors** $x_i \in X$ **such that** $A(x_i) = 0$:

$$\mathrm{N}(A) = \left\{ x \in X \middle| A(x) = 0 \right\}$$

For linear operator *A*, the range space $\mathcal{R}(A)$ is a *subspace* of *Y (its dimension is the rank of matrix A)*, and the null space $\mathcal{N}(A)$ is a *subspace* of *X (its dimension is the nullity of matrix A)*.

How to represent Linear Operators: We will now show that a linear operator on a vector space can always be written in the form of a matrix.

Consider the vectors *x* and *y* from n- and m- dimensional spaces:

$$x \in X^n, \quad y \in X^m.$$

Let $\{v_1, \quad v_2, \quad \dots, \quad v_n\}$ be a basis for X^n

and $\{u_1, \quad u_2, \quad \dots, \quad u_m\}$ be a basis for X^m

Then by expanding *x* out in its basis vectors, $x = \sum_{j=1}^{n} \alpha_j v_j$ **so**

$$y = A(x) = A\left(\sum_{j=1}^{n} \alpha_j v_j\right) = \sum_{j=1}^{n} A(\alpha_j v_j) = \sum_{j=1}^{n} \alpha_j A(v_j)$$

This important results means that we can determine the effect of *A* on any vector *x* by knowing only the effect of *A* on the basis vectors of X^n **.**

We can write this out in matrix-vector notation as:

$$y = A(x) = \left[A(v_1) \;\vdots\; A(v_2) \;\vdots\; \cdots \;\vdots\; A(v_n)\right] \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}$$

Now we note that each $A(v_j)$ is itself a vector in the space X^m so it, like *y* itself, can always be expanded as a unique representation in the basis $\{u_i\}$ of X^m:

$$A(v_j) = \sum_{i=1}^{m} a_{ij} u_i \text{ and } y = \sum_{i=1}^{m} \beta_i u_i$$

Substituting these into $y = \sum_{j=1}^{n} \alpha_j A(v_j)$ from the previous page,

$$y = \sum_{j=1}^{n} \alpha_j \left[\sum_{i=1}^{m} a_{ij} u_i \right]$$

And changing the order of summation:

$$y = \sum_{i=1}^{m}\left[\sum_{j=1}^{n} a_{ij}\alpha_j\right]u_i = \sum_{i=1}^{m}\beta_i u_i$$

But the expansion of *y* into $\{u_i\}$ must be unique, so

$$\boxed{\beta_i = \sum_{j=1}^{n} a_{ij}\alpha_j \quad \text{for all } \; i = 1, \ldots, m}$$

Operator

$b = Aa$

1st basis

What good does all this do us????

Suppose we have the representations

$$x \in X^n = \alpha = [\alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_n] \text{ in the } \{v_j\} \text{ basis, and}$$

$$y \in X^m = \beta = [\beta_1 \quad \beta_2 \quad \cdots \quad \beta_m] \text{ in the } \{u_i\} \text{ basis}$$

If $y = A(x)$, we can find the components β_i of *y* from the formula above.

This can be written as a multiplication of the *n*-dimensional vector a by the (m x n) matrix $A = [a_{ij}]$ to get the *m*-dimensional vector b .

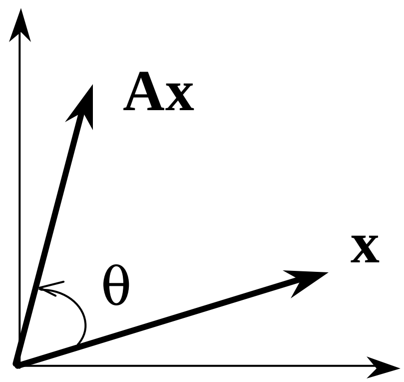
This *matrix A* is the matrix representation of *transformation A*. The elements of this matrix obviously are going to depend on that particular bases chosen, so unless it is clear from the context, one must always specify the basis in X^n and the basis in X^m.

"A" Transforms the input basis space to the Range basis space.

Careful examination of the subscripts in the expression shows that the *j* th column of *A* is the representation of its effect on the *j* th basis vector in the domain, expanded into the range basis. That is, the *j* th column of *A* is $A(v_j)$ written out in the basis. $\{u_i\}$

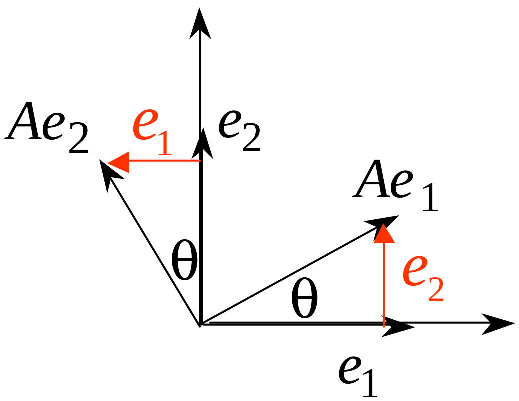
A *very* useful property, as we shall now see in some examples.

<u>EXAMPLE</u>: Consider the linear vector space of all vectors in 2-D (R^2). We can define a rotation operator as the linear transformation *A* that rotates a vector *x* by an angle q counterclockwise.



Find the matrix representation for A.

Recall the rule that says that the *i* th column of *A* is the effect of *A* on the *i* th basis vector:



Effect of A on e_1 :

$$Ae_1 = e_1 \cos\theta + e_2 \sin\theta = [e_1 \quad e_2]\begin{bmatrix}\cos\theta \\ \sin\theta\end{bmatrix}$$

Effect of A on e_2 :

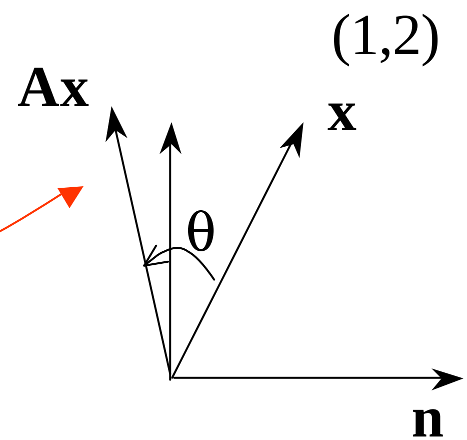
$$Ae_2 = -e_1 \sin\theta + e_2 \cos\theta = [e_1 \quad e_2]\begin{bmatrix}-\sin\theta \\ \cos\theta\end{bmatrix}$$

So

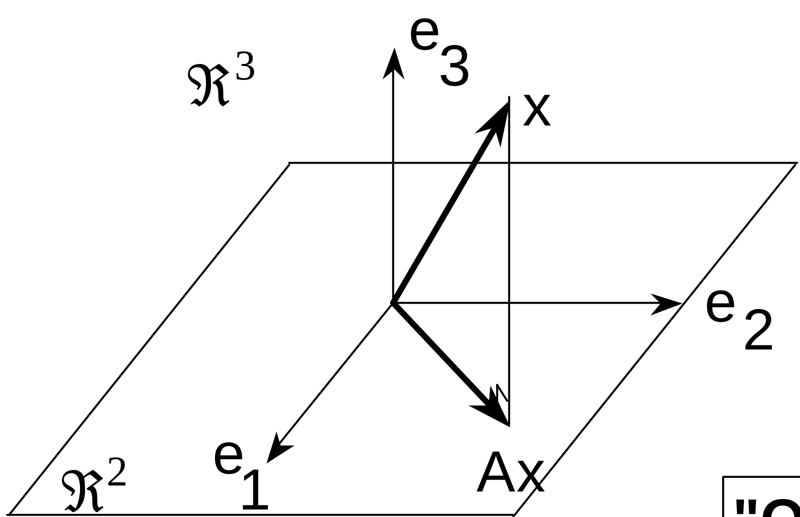
$$A = \begin{bmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{bmatrix}$$

Try this on vector $x = \begin{bmatrix}1 \\ 2\end{bmatrix}$; let θ be 30° :

$$Ax = \begin{bmatrix}\cos(30) & -\sin(30) \\ \sin(30) & \cos(30)\end{bmatrix}\begin{bmatrix}1 \\ 2\end{bmatrix} = \begin{bmatrix}-.134 \\ 2.23\end{bmatrix}$$



<u>EXAMPLE</u>: Let A be the linear operator that forms an *orthogonal projection* from a 3-D space into a 2-D space. Suppose the 2-D space is the "x-y plane" of the 3-D space:



$$A: \Re^3 \to \Re^2$$

"Orthogonal" projection is along the z-axis. We could also project "along" any other line, but this wouldn't be *orthogonal*.

We can form the matrix representation for *A* from the effect it has on the basis vectors. Let the basis for R^2 be $\{e_1, e_2\}$ and the basis for R^3 is $\{e_1, e_2, e_3\}$.

$$Ae_1 = e_1 = [e_1 \quad e_2]\begin{bmatrix}1\\0\end{bmatrix}$$

$$Ae_2 = e_2 = [e_1 \quad e_2]\begin{bmatrix}0\\1\end{bmatrix}$$

$$\left. Ae_3 = 0 = [e_1 \quad e_2]\begin{bmatrix}0\\0\end{bmatrix} \right\} \text{zero out}$$

So

$$\boxed{A = \begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\end{bmatrix}}$$

a (2 x 3) matrix

■

We often see transformations from a space *X* into *itself* (which results in a square matrix representation of *A*). It is possible for this transformation to map representations from one basis into representations in a different basis, but we'll seldom find use for these.

Suppose we consider a linear transformation that maps vectors from space *X* into *itself*: What happens when the *basis* is different?

For notation, let $\hat{A}$ be the transformation in the basis $\{\hat{v}_i\}$, and
A be the transformation in the basis $\{v_i\}$; i.e.,

$$[y]_v = A[x]_v \quad \text{and} \quad [y]_{\hat{v}} = \hat{A}[x]_{\hat{v}}$$

Now because we're working all within a single space, we can use a *B*-matrix to transform both *x* and *y* vectors from ”hat" to ”no hat" coordinates:

$$\left.\begin{aligned} [y]_{\hat{v}} &= \hat{A}[x]_{\hat{v}} \\ [B][y]_{v} &= \hat{A}[B][x]_{v}, \text{ so} \\ [y]_{v} &= [B]^{-1}\hat{A}[B][x]_{v} \end{aligned}\right\} \quad \begin{aligned} [y]_{\hat{v}} &= B[x]_{v} \\ [x]_{\hat{v}} &= B[x]_{v} \end{aligned}$$

Comparing to $[y]_{v} = A[x]_{v}$ **from the previous page, we see that**

$$\boxed{A = B^{-1}\hat{A}B}$$

And if the bases are both orthonormal, the inverse is equal to the transpose, so

$$\boxed{A = B^{T}\hat{A}B}$$

This is how we change the expression of a linear transformation *A* from one basis into another basis, and it is called a similarity transformation.

<u>EXAMPLE</u>: Consider the linear vector space of all polynomials in s, of degree less than 4, with constant coefficients (over the field of reals). One can show that the operator *A* that takes a vector *v(s)* and transforms it into

$$v''(s) + 2v'(s) + 3v(s)$$

is *linear operator* from the space into itself. Find its matrix representation if the basis is:

$$\{e_i\} = \{s^3 \quad s^2 \quad s \quad 1\}$$

Find the effect on the basis vectors:

$$Ae_1 = 6s + 6s^2 + 3s^3 = [e_1 \quad e_2 \quad e_3 \quad e_4]\begin{bmatrix}3\\6\\6\\0\end{bmatrix}$$

<u>Operator</u>

$$V''(s) + 2V'(s) + 3V(s)$$

$$\{e_i\} = \{s^3 \quad s^2 \quad s \quad 1\}$$

derivative s :

1^{st} : $\{3s^2 \quad 2s \quad 1 \quad 0\}$

2^{nd} : $\{6s \quad 2 \quad 0 \quad 0\}$

$$Ae_2 = 2 + 4s + 3s^2 = [e_1 \quad e_2 \quad e_3 \quad e_4]\begin{bmatrix}0\\3\\4\\2\end{bmatrix}$$

$$e_1 = s^3, e_2 = s^2, e_3 = s, e_4 = 1$$

$$Ae_3 = 2 + 3s = [e_1 \quad e_2 \quad e_3 \quad e_4]\begin{bmatrix}0\\0\\3\\2\end{bmatrix}$$

So $$\underbrace{A = \begin{bmatrix}3 & 0 & 0 & 0\\6 & 3 & 0 & 0\\6 & 4 & 3 & 0\\0 & 2 & 2 & 3\end{bmatrix}}_{\text{operator}}$$

$$Ae_4 = 3 = [e_1 \quad e_2 \quad e_3 \quad e_4]\begin{bmatrix}0\\0\\0\\3\end{bmatrix}$$

Try this out on the vector

$$v(s) = s^2 + 1, \quad \text{whose } \textit{representation} \text{ is } v = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{matrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{matrix}$$

$$Av = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 6 & 3 & 0 & 0 \\ 6 & 4 & 3 & 0 \\ 0 & 2 & 2 & 3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

$$v(s) = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} s^3 & s^2 & s & 1 \end{bmatrix}^T$$

Check: $v''(s) + 2v'(s) + 3v(s) = 3s^2 + 4s + 5 \Big\}$ Plug in v(s)

Now, how would this linear operator be represented if the basis were instead:

$$\{\bar{e}_i\} = \{s^3 - s^2, \quad s^2 - s, \quad s - 1, \quad 1\} \Big\}$$ How are the basis related ?

To do this, we need that *B*-matrix that contains the coefficients that relate the new and old bases. Recall the relation:

$$e_j = \sum_{i=1}^{n} b_{ij} \bar{e}_i \quad \left.\begin{matrix} \\ \\ \end{matrix}\right\} \begin{matrix} \text{Defines B} \\ e = B\bar{e} \end{matrix}$$

It is easier to see the *inverse* relationship in this case:

$$\left.\begin{aligned}
\bar{e}_1 &= e_1 - e_2 = \begin{bmatrix} e_1 & e_2 & e_3 & e_4 \end{bmatrix}\begin{bmatrix} 1 & -1 & 0 & 0 \end{bmatrix}^T \\
\bar{e}_2 &= e_2 - e_3 = \begin{bmatrix} e_1 & e_2 & e_3 & e_4 \end{bmatrix}\begin{bmatrix} 0 & 1 & -1 & 0 \end{bmatrix}^T \\
\bar{e}_3 &= e_3 - e_4 = \begin{bmatrix} e_1 & e_2 & e_3 & e_4 \end{bmatrix}\begin{bmatrix} 0 & 0 & 1 & -1 \end{bmatrix}^T \\
\bar{e}_4 &= e_4 = \begin{bmatrix} e_1 & e_2 & e_3 & e_4 \end{bmatrix}\underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T}_{\text{forms } B^{-1}}
\end{aligned}\right\} \text{Some Tinkering}$$

Which of course gives us B^{-1} instead of B:

$$B^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \text{ from which we compute } B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Now from the formula: $A = B^{-1}\bar{A}\,B$

$\boxed{e = B\bar{e}}$ } Basis change defined sets the order

we solve for $\bar{A}$: $\bar{A} = BAB^{-1}$ **and compute:**

$$\boxed{\bar{A} = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 6 & 3 & 0 & 0 \\ 8 & 4 & 3 & 0 \\ 6 & 4 & 2 & 3 \end{bmatrix}}$$

How do we check this? First find the representation of our vector *v* in the new basis:

By definition $\boxed{\bar{v} = Bv} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 2 \end{bmatrix}$

$$\left(= 1\bar{e}_2 + 1\bar{e}_3 + 2\bar{e}_4 = (s^2 - s) + (s-1) + 2 = s^2 + 1\right)$$

Now compute the differential operation within this basis:

substitute $s^2 + 1$

$v'' + 2v' + 3v$

$$\bar{A}\bar{v} = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 6 & 3 & 0 & 0 \\ 8 & 4 & 3 & 0 \\ 6 & 4 & 2 & 3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 7 \\ 12 \end{bmatrix}$$

☺

$$\left(= 3\bar{e}_2 + 7\bar{e}_3 + 12\bar{e}_4 = 3(s^2 - s) + 7(s-1) + 12(1) = 3s^2 + 4s + 5\right)$$

■

Operations on Operators:

Linear transformations, like the matrices that represent them, are generally not commutative, but they can be added. That is, to find

$$y = A_1(x) + A_2(x),$$

we can add the matrix representations for A_1 and A_2 **and compute:**

$$(A_1 + A_2)x = y$$

Operator Norms:

Sometimes we want to know how big *A* will make *x*. We then give the operator *A* a norm:

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}, \text{ or equivalently } \quad \|A\| = \sup_{\|x\|=1} \|Ax\|$$

Recall that there are many ways to define $\|x\|$. Consequently, there are many different matrix norms. They all follow the rules:

$$\|Ax\| \le \|A\| \cdot \|x\| \text{ for all } x$$

$$\|A_1 + A_2\| \le \|A_1\| + \|A_2\|$$

$$\|A_1 A_2\| \le \|A_1\| \cdot \|A_2\|$$

$$\|\alpha A\| = |\alpha| \cdot \|A\|$$

EXAMPLE: The 2-norm of a matrix can be written as

$$\|A\| = \left[\max_{\|x=1\|} \left\{ \overline{x}^T \, \overline{A}^T A x \right\} \right]^{1/2}$$

Note that this definition does not indicate how to *compute* such a quantity!!

And finally,

<u>DEFINITION</u>: The *adjoint* of the linear operator A is denoted A* and is defined by

$$\langle Ax\ , y\rangle = \langle x, A^* y\rangle$$

for all *x* and *y*. *A is itself a linear operator, and for matrix representations, happens to be the complex conjugate transpose of *A*:**

$$A^* = \overline{A}^T$$

Simultaneous Linear Equations

Consider the familiar set of *m* linear equations in *n* unknowns:

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = y_1$$
$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = y_2$$
$$\vdots$$
$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = y_m$$

Write this in matrix-vector form:

$$Ax = y$$

m* x *n	***n* x 1**	***m* x 1**
$(A: X^n \to X^m)$	$(x \in X^n)$	$(y \in X^m)$

We wish to investigate the circumstances of when, if and how many solutions exists to this matrix equation.

Consider that *x* is a vector of coefficients of a linear combination of the columns of *A*.

$$Ax = \begin{bmatrix} \\ a_1 \\ \\ \end{bmatrix} x_1 + \begin{bmatrix} \\ a_2 \\ \\ \end{bmatrix} x_2 + \cdots + \begin{bmatrix} \\ a_n \\ \\ \end{bmatrix} x_n = \begin{bmatrix} \\ y \\ \\ \end{bmatrix}$$

For the equality to hold, then, *y* would have to be equal to a linear combination of the columns of A ("in the *column space* of matrix A."

If this is the case, then *y* will be linearly dependent on the columns of *A*.

...So appending the vector *y* to the columns from *A* cannot add any rank to those columns, considered as a set of vectors. That is, if

$$W = \left[A \,\vdots\, y \right] \quad \textbf{then} \quad r(A) = r(W)$$

$r(A) = r(W)$ **when at least* one solution exists.**

$\left(y \in \mathbf{R}(A)\right)$

Conversely

$r(A) \neq r(W)$

When NO solutions exists.

* **If** $r(A) = r(W)$**, we can have either a *unique* solution, or possibly an *infinite* number of solutions.**

If $r(A) = r(W) = n,$ then the solution x is unique,

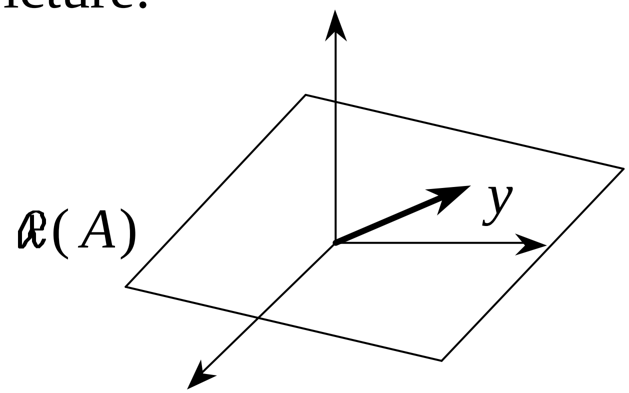
If $r(A) = r(W) < n,$ then there are an infinite number of solutions.

Why?

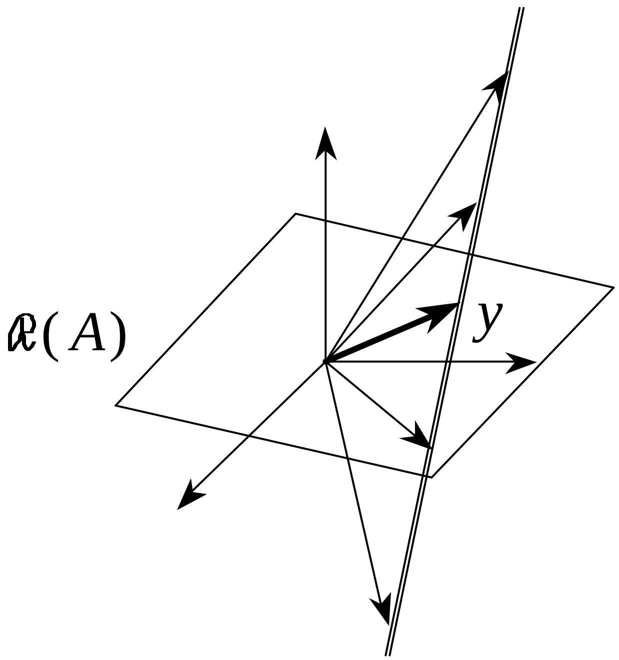
If $r(A) = n$**, then the columns of *A* form a basis of the n-dimensional column space of *A*. Then vectors within this space (of which *y*, we have said, is one, because** $r(W) = r(A)$ **) will be written as *unique* linear combinations of the basis vectors.**

If $r(A) < n$**, then the columns of *A* still span the column space, but there are too many of them to be a basis, so the *y* vector is not a unique linear combination of the columns of *A*, but one of an infinite number of possibilities.**

A Picture:



Suppose $r(A) = 2$ **, depicted by the plane. Then since all products *Ax* lie in that plane, *y* had better be there, or else there is no *x* such that *Ax*=*y*.**



Now suppose *n*=3; i.e., *A* maps 3-D vectors into the 2-D plane. Then there are lots of *x*'s that might map to the 2-D vector *y*.

$$r(A) < n$$

What's the easiest way to find out what $r(A)$ and $r(W)$ are?

$\longrightarrow$ **computer!!**

By hand? $\longrightarrow$ **"Echelon Form"**

(resulting from "elementary operations")

Many examples in book. $W = [A \;\vdots\; Y]$

notation: W' **=echelon form of** W

<u>Ex. 3:</u>

$$W = \left[\begin{array}{ccc:c} 1 & 2 & 3 & 4 \\ 2 & 1 & 2 & 7 \\ 3 & 2 & 1 & 1 \end{array}\right] \qquad W' = \left[\begin{array}{ccc:c} 1 & 0 & 0 & 2.125 \\ 0 & 1 & 0 & -4.5 \\ 0 & 0 & 1 & 3.625 \end{array}\right]$$

This is it!

$r(A) = r(W) = 3 = n$

Unique solution exists

Ex. 4:

$$W = \left[\begin{array}{ccc:c} 1 & -1 & 2 & 8 \\ -1 & 2 & 0 & 2 \end{array}\right] \qquad W' = \left[\begin{array}{ccc:c} 1 & 0 & 4 & 18 \\ 0 & 1 & 2 & 10 \end{array}\right]$$

$r(A) = r(W) = 2 < n$ **An infinite number of solutions exists, and must satisfy** $x_1 + 4x_3 = 18$

$$x_2 + 2x_3 = 10$$

Ex. 5:

$$W = \left[\begin{array}{cc:c} 1 & 2 & 2 \\ 3 & 4 & 3 \\ 5 & 6 & -4 \end{array}\right] \qquad W' = \left[\begin{array}{cc:c} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right]$$

$r(A) = 2, \ r(W) = 3$ **NO solutions exist!**

(Note here that $m > n$ **)**

So how do we actually find the solutions, when they exist? $(r(A) = r(W))$

Echelon form goes a long way, but is manual labor.

If A is $n \times n$ and $r(A) = r(W) = n$, **Then we just invert** $\boldsymbol{A}$**:**

$$x = A^{-1}y$$

If not . . .

Two common cases: "Overdetermined" $(m > n)$

and "Underdetermined" $(m < n)$

<u>Underdetermined Case:</u> $m < n$

There is no possibility of a unique solution because

$$r(A) \leq (\min(n,m) = m) < n$$

Usually, when we have a choice of many solutions, we prefer the smallest one. That is, we will choose the *x* with *minimum norm*, or equivalently, minimum *squared-norm*.

That, we will *minimize* $\frac{1}{2} x^T x$

Subject to a *constraint* that: $y = Ax$

We'll do this using a "*LaGrange Multiplier*"

Find: $\min_x (\tfrac{1}{2} x^T x)$ such that $Ax - y = 0$

Form "Hamiltonian" $H = \tfrac{1}{2} x^T x + \lambda^T (y - Ax)$

LaGrange multiplier; a vector that makes our constraint equation the same dimension as the scalar we are minimizing. λ is actually part of our unknown, so we take our derivatives w.r.t it, too:

$$\frac{\partial^{\mathrm{T}} H}{\partial x} = x - A^T \lambda = 0$$

$$\frac{\partial^{\mathrm{T}} H}{\partial \lambda} = y - Ax = 0$$

Note that this is the same as our "constraint" equation!

(repeating)

$$\frac{\partial^{\mathrm{T}} H}{\partial x} = x - A^T\lambda = 0 \qquad \textbf{(1)}$$

$$\frac{\partial^{\mathrm{T}} H}{\partial \lambda} = y - Ax = 0 \qquad \textbf{(2)}$$

Multiply eqn. (1) by *A*: $Ax = AA^T\lambda$

But eqn. (2) gives $y = Ax = AA^T\lambda$

A must be full rank

So $\lambda = (AA^T)^{-1} y$

How do we know $(AA^T)^{-1}$ **exists?**

Substitute this back into (1) to get $\boxed{x = A^T(AA^T)^{-1} y}$

Sometimes called a "pseudoinverse" of *A*.

$$x = A^T (AA^T)^{-1} y$$

Is the "shortest" vector such that $y = Ax$
for a given *y* and *A*.

"Overdetermined Case" $m > n$

If $r(A) = n$**, then we know there will be *one* solution.**

However it often happens that there is no solution, because *y* has *m* elements, and we are asking that it be a linear combination of only *n* vectors $(n < m)$**. This often happens even when we know there must be a solution, because, e.g., we are taking lots of data (equations) from an experiment with only a few variables (*x*'s).**

In this situation, the noisy data might give $r(A) \neq r(W)$ **so we settle for the** ***x*** **that gives us the smallest** ***error*** **in the equation** $Ax = y$.

Whether a single solution exists or we want an approximate solution with smallest error, the following procedure works:

Define error: $e = y - Ax$

Now find: $\min_x \frac{1}{2}\|e\|^2 = \min_x \frac{1}{2} e^T e$

$$\frac{1}{2}e^T e = \frac{1}{2}\left[(y - Ax)^T (y - Ax)\right]$$
$$= \frac{1}{2}\left[(y^T - x^T A^T)(y - Ax)\right]$$

$$\tfrac{1}{2} e^T e = \tfrac{1}{2}\left[(y^T - x^T A^T)(y - Ax)\right]$$

$$= \tfrac{1}{2}\left[y^T y - x^T A^T y - y^T Ax + x^T A^T Ax\right]$$

These are equal because they are scalars and transposes of each other!

$$\tfrac{1}{2} e^T e = \tfrac{1}{2}\left[y^T y - 2x^T A^T y + x^T A^T Ax\right]$$

Take derivative and set = to zero:

$$\frac{\partial^{\mathrm{T}}\left[\tfrac{1}{2} e^T e\right]}{\partial x} = \tfrac{1}{2}\left[-2A^T y + 2A^T Ax\right] = 0$$

Solving:

$$\boxed{x = (A^T A)^{-1} A^T y}$$

$$x = (A^T A)^{-1} A^T y$$

Note the striking similarity to the "pseudoinverse" we saw in the underdetermined case $x = A^T (AA^T)^{-1} y$

This is sometimes also called a pseudoinverse. (a different one; pseudoinverses are not unique, they also come in left- and right- versions).

<u>Example:</u> **Suppose**

$$A = \begin{bmatrix} 2 & 2 \\ 1 & 2 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad y = \begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix}$$

Find x such that $y = Ax$ **, or** ***x*** **such that** $y - Ax$ **is as small as possible if no exact solution exists.**

$$A = \begin{bmatrix} 2 & 2 \\ 1 & 2 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad y = \begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix}$$

$$(A^T A)^{-1} A^T y = \begin{bmatrix} 1/3 & -1/3 & 2/3 \\ 0 & 1/2 & -1/2 \end{bmatrix} \begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 1 \end{bmatrix} = x$$

To see how good an approximation this is, compute the error vector:

$$e = y - Ax = \begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix} - \begin{bmatrix} 2 & 2 \\ 1 & 2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

So there was no error at all. We must have $y \in \mathcal{R}(A)$ **(which may have been obvious anyway).**

We had $r(A) = r(W) = n$ **so the solution existed and was unique!**

R=2

Two Important Examples in Control Systems:

Consider a *discrete-time* system in state-space form:

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k) + Du(k)$$

Do some brute-force recursive calculations:

$$\boxed{\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned}}$$

$$\begin{aligned} x(1) &= Ax(0) + Bu(0) \\ y(0) &= Cx(0) + Du(0) \end{aligned}$$

$$\begin{aligned} x(2) &= Ax(1) + Bu(1) = A^2x(0) + ABu(0) + Bu(1) \\ y(1) &= Cx(1) + Du(1) = CAx(0) + CBu(0) + Du(1) \end{aligned}$$

$$\begin{aligned} x(3) &= A^3x(0) + A^2Bu(0) + ABu(1) + Bu(2) \\ y(2) &= CA^2x(0) + CABu(0) + CBu(1) + Du(2) \end{aligned}$$

$$\vdots$$

make note of these patterns forming

etc.

<u>Problem #1:</u> When is it possible to make $x(k) = 0$ by applying the sequence $u(0), \ldots, u(k-1)$ regardless of what $x(0)$ is?

Consider the equation:

$$0 = x(k) = A^k x(0) + A^{k-1} B u(0) + A^{k-2} B u(1) + \ldots + B u(k-1)$$

(from the pattern on the previous page)

Re-arrange:

$$-A^k x(0) = \underbrace{\left[B \;\vdots\; \cdots \;\vdots\; A^{k-2} B \;\vdots\; A^{k-1} B \right]}_{\triangleq\, P} \begin{bmatrix} u(k-1) \\ \vdots \\ u(1) \\ u(0) \end{bmatrix}$$

***n* x 1**

We are allowing $x(0)$ to be *any* n-dimensional vector, so by our knowledge of linear equations, we want to have $\boxed{r(P) \geq n}$. (We will find out later that the *P* matrix cannot have rank *greater* than *n*). Systems with this property are called *controllable*.

<u>Problem #2:</u> If at time *k*, we know the *current* and all the *previous*, inputs and outputs, then when will it be possible to figure out what $x(0)$ was?

Recall the recursion equations:

$$y(0) = Cx(0) + Du(0)$$

$$y(1) = CAx(0) + CBu(0) + Du(1)$$

$$y(2) = CA^2x(0) + CABu(0) + CBu(1) + Du(2)$$

$$\vdots$$

etc.

Re-arrange these as:

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(k) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^k \end{bmatrix} x(0) + \boxed{\begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & \cdots & 0 \\ \vdots & & & \ddots & \\ CA^{k-1}B & & & \cdots & D \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ \vdots \\ u(k) \end{bmatrix}}$$

combine with other side

Re-write again as:

known

$$\Psi_k = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^k \end{bmatrix} x(0) \stackrel{\Delta}{=} Qx(0)$$

Similar to before, this Q -matrix is going to need to be full (n) rank in order for the linear equation above to have a solution. Then knowing the left-hand side, which contains the past inputs and outputs, we can find an arbitrary initial condition!

This system is *observable*; a concept we'll see again soon.