Math 314/814

Topics for first exam

Chapter 2: Systems of linear equations

Some examples

Systems of linear equations:

$$2x - 3y - z = 6$$
$$3x + 2y + z = 7$$

Goal: find simultaneous solutions: all x, y, z satisfying both equations.

Most general type of system:

$$a_{11}x_1 + \cdots + a_{1n}x_n = b_1$$
$$\cdots$$
$$a_{m1}x_1 + \cdots + a_{mn}x_n = b_m$$

Gaussian elimination: basic ideas

$$3x + 5y = 2$$
$$2x + 3y = 1$$

Idea use $3x$ in first equation to eliminate $2x$ in second equation. How? Add a multiply of first equation to second. Then use y-term in new second equation to remove $5y$ from first!

The point: a solution to the original equations **must** also solve the new equations. The **real** point: it's much easier to figure out the solutions of the new equations!

Streamlining: keep only the essential information; throw away unneeded symbols!

$$\begin{array}{lll} \text{3x+5y=2} & \text{replace} & \left(\begin{array}{cc|c} 3 & 5 & 2 \\ 2 & 3 & 1 \end{array}\right) \\ \text{2x+3y=1} & \text{with} & \end{array}$$

We get an **(augmented) matrix** representing the system of equations. We carry out the same operations we used with equations, but do them to the rows of the matrix.

Three basic operations (elementary row operations):

E_{ij} : switch ith and jth rows of the matrix

$E_{ij}(m)$: add m times jth row to the ith row

$E_i(m)$: multiply ith row by m

Terminology: first non-zero entry of a row = **leading entry**; leading entry used to zero out a column = **pivot**.

Basic procedure (Gauss-Jordan elimination): find non-zero entry in first column, switch up to first row (E_{1j}) (pivot in (1,1) position). Use $E_1(m)$ to make first entry a 1, then use $E_{1j}(m)$ operations to zero out the other entries of the first column. Then: find leftmost entry in remaining rows, switch to second row, use as a pivot to clear out the entries in the column below it. Continue (forward solving). When done, use pivots to clear out entries in column above the pivots (back-solving).

Variable in linear system corresponding to a pivot = **bound** variable; other variables = **free** variables

Gaussian elimination: general procedure

The big fact: After elimination, the new system of linear equations have the exact **same solutions** as the old system. Because: row operations are reversible!

Reverse of E_{ij} is E_{ij}; reverse of $E_{ij}(m)$ is $E_{ij}(-m)$; reverse of $E_i(m)$ is $E_i(1/m)$

So: you can get old equations from new ones; so solution to new equations **must** solve old equations **as well**.

Reduced row form: apply elementary row operations so turn matrix A into one so that

(a) each row looks like $(000\cdots 0 * * \cdots *)$; firsdt $*$ = leading entry

(b) leading entry for row below is further to the right

Reduced row **echelon** form: in addition, have

(c) each leading entry is $= 1$

(d) each leading entry is the only non-zero number in its column.

RRF can be achieved by forward solving; RREF by back-solving and $E_i(m)$'s
Elimination: every matrix can be put into RREF by elementary row operations.
Big Fact: If a matrix A is put into RREF by two different sets of row operations, you get the **same matrix**.
RREF of an augmented matrix: can read off solutions to linear system.

$$\left(\begin{array}{cccc|c} 1 & 0 & 2 & 0 & 2 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 3 \end{array}\right) \quad \text{means} \quad \begin{array}{l} x_4=3,\ x_2=1-x_3 \\ x_1=2-2x_3 \quad ; x_3 \text{ is free} \end{array}$$

Inconsistent systems: row of zeros in coefficient matrix, followed by a non-zero number (e.g., 2). Translates as 0=2 ! System has no solutions.

Rank of a matrix = r(A) = number of non-zero rows in RREF = number of pivots in RREF.
Nullity of a matrix = n(A) = number of columns without a pivot = # columns − # pivots
rank = number of bound variables, nullity = number of free variables
rank $\leq$ number of rows, number of columns (at most one pivot per row/column!)
rank + nullity = number of columns = number of variables

A = coefficient matrix, $\tilde{A}$ = augmented matrix ($A = m \times n$ matrix)
system is consistent if and only if $\mathrm{r}(A) = \mathrm{r}(\tilde{A})$
$\mathrm{r}(A)=n$: unique solution ; $\mathrm{r}(A)< n$: infinitely many solutions

Spanning sets and linear independence

We can interpret an SLE in terms of (column) vectors; writing v_i = ith column of the coefficient matrix, and b=the column of target values, then our SLE really reads as a single equation $x_1v_1 + \cdots + x_nv_n = b$. The lefthand side of this equation is a *linear combination* of the vectors $v_1, \ldots, v_n$, that is, a sum of scalar multiples. Asking if the SLE has a solution is the same as asking if b is a linear combination of the v_i.
This is an important enough concept that we introduce new terminology for it; the *span* of a collection of vectors, $\mathrm{span}(v_1, \ldots, v_n)$ is the collection of <u>all</u> linear combinations of the vectors. If the span of the $(m \times 1)$ column vectors $v_1, \ldots, v_n$ is all of $\mathbb{R}^m$, we say that the vectors span $\mathbb{R}^m$. Asking if an SLE has a solution is the same as asking if the target vector is in the span of the column vectors of the coefficient matrix.

The flipside of spanning is *linear independence*. A collection of vectors $v_1, \ldots, v_n$ is linearly independent if the only solution to $x_1v_1 + \cdots + x_nv_n = 0$ (the 0-vector) is $x_1 = \cdots = x_n = 0$ (the "trivial" solution). If there is a non-trivial solution, then we say that the vectors are linearly dependent. If a collection of vectors is linearly dependent, then choosing a non-trivial solution and a vector with non-zero coefficient, throwing everything else on the other side of the equation expresses one vector as a linear combination of all of the others. Thinking in terms of an SLE, the columns of a matrix are linearly dependent exactly when the SLE with target 0 has a non-trivial solution, i.e., has more than one solution. It has the trivial (all 0) solution, so it is consistent, so to have more than one, we need the the RREF for the matrix to have a free variable, i.e., the rank of the coefficient matrix is less than the number of columns.

Some applications

Allocation of resources:
If we have a collection of objects to manufacture, each requiring known amounts of the same collection of resources, then manufacturing differing amounts of the objects will use differing amounts of the resources. If we have a fixed amount of each resource, we can determine how much of each object to make in order to use all of our resources, by creating a system of linear equations. We treat the amounts of each object we will manufacture as a vector of unknown variables $\vec{v} = (x, y, z, \dots)^T$, and we assemble the amounts of each resource $A, B, C, \dots$ needed to manufacture each object as a matrix M, with objects to manufacture indexing the columns, and the resources indexing the rows. Then an "A" row reading (2 3 1 4) means that in order to constuct (x, y, z, w) units of the objects we need $2x + 3y + z + 4w$ units of the resource "A". [Columns of this matrix therefore represent the amounts of each resource needed to manufacture one unit of the object indexed by that column.] So if we know the amount of each resource on hand, assembled as a column vector $\vec{b}$, then we can solve our resource allocation problem, namely, "what to manufacture in order to use all of our available resources?", by solving the system of linear equations $M\vec{v} = \vec{b}$.

Balancing chemical reactions:
In a chemical reaction, some collection of molecules is converted into some other collection of molecules. The proportions of each can be determined by solving an SLE:

E.g., when ethane is burned, $x\ C_2H_6$ and $y\ O_2$ is converted into $z\ CO_2$ and $w\ H_2O$. Since the number of each element must be the same on both sides of the reaction, we get a system of equations

$C : 2x = z$; $H : 6x = 2w$; $O : 2y = w$.
which we can solve. More complicated reactions, e.g., $PbO_2 + HCl \to PbCl_2 + CL_2 + H_2O$, yield more complicated equations, but can still be solved using the techniques we have developed.

Network Flow:
We can model a network of water pipes, or trafic flowing in a city's streets, as a graph, that is, a collection of points = vertices (=intersections=junctions) joined by edges = segments (=streets = pipes). Monitoring the flow along particular edges can enable us to know the flow on every edge, by solving a system of equations; at every vertex, the net flow must be zero. That is, the total flow into the vertex must equal the total flow out of the vertex. Giving the edges arrows, depicting the direction we think traffic is flowing along that edge, and labeling each edge with either the flow we know (monitored edge) or a variable denoting the flow we don't, we have a system of equations (sum of flows into a vertex) = (sum of flows out of the vertex). Solving this system enables us to determine the value of every variable, i.e., the flow along every edge. A negative value means that the flow is opposite to the one we expected.

Chapter 3: Matrices

Matrix addition and scalar multiplication

Idea: take our ideas from vectors. Add entry by entry. Constant multiple of matrix: multiply entry by entry.
$\mathbf{0}$ = matrix all of whose entries are 0
Basic facts:

A+B makes sense only if A and B are the same size (m×n) matrix
A+B = B+A
(A+B)+C = A+(B+C)
A+$\mathbf{0}$ = A
A+(-1)A = $\mathbf{0}$

cA has the same size as A
c(dA) = (cd)A
(c+d)A = cA + dA
c(A+B) = cA + cB
1A = A

Matrix multiplication

Idea: **don't** multiply entry by entry! We want matrix multplication to allow us to write a system of linear equations as Ax=b ....
Basic step: a row of A, times x, equals an **entry** of Ax. (row vector $(a_1, \ldots, a_n)$ times column vector $(x_1, \ldots, x_n)$ is $a_1x_1 + \cdots + a_nx_n$ ....) This leads to:
In AB, each row of A is 'multiplied' by each column of B to obtain an entry of AB. Need: the length of the rows of A (= number of columns of A) = length of columns of B (= number of rows of B). I.e, in order to multiply, A must be m×n, and B must be n×k; AB is then m×k.

Formula: (i,j)th entry of AB is $\Sigma_{k=1}^{n} a_{ik}b_{kj}$

I = identity matrix; square matrix (n×n) with 1's on diagonal, 0's off diagonal
Basic facts:
AI = A = IA
(AB)C = A(BC)
c(AB) = (cA)B = A(cB)
(A+B)C = AC + BC
A(B+C) = AB + AC

In general, however it is ****not** **not**** true that AB and BA are the same; they are almost always different! ********

Special matrices and transposes

Elementary matrices:
A row operation (E_{ij} , $E_{ij}(m)$, $E_i(m)$) applied to a matrix A corresponds to multiplication (on the left) by a matrix (also denoted E_{ij} , $E_{ij}(m)$, $E_i(m)$) The matrices are obtained by applying the row operation to the identity matrix I_n. E.g., the 4×4 matrix $E_{13}(-2)$ looks like I, except it has a -2 in the (1,3)th entry.
The idea: if A → B by the elementary row operation E, then B = EA.
So if A → B → C by elementary row operations, then C = E_2E_1A ....
Row reduction **is** matrix multiplication!

A scalar matrix A has the same number c in the diagonal entries, and 0's everywhere else (the idea: AB = cB)
A diagonal matrix has all entries zero off of the (main) diangonal
A upper triangular matrix has entries =0 below the diagonal, a lower triangular matrix is 0 above the diagonal. A triangular matrix is either upper or lower triangular.
A strictly triangular matrix is triangular, and has zeros **on** the diagonal, as well. They come in upper and lower flavors.

The **transpose** of a matrix A is the matrix A^T whose columns are the rows of A (and vice versa). A^T is A reflected across the main diagonal. $(aij)^T = (aji)$; (m×n)^T = (n×m)
Basic facts:
$(A+B)^T = A^T + B^T$
$(AB)^T = B^TA^T$
$(cA)^T =$ cA^T

$(A^T)^T$ = A
Transpose of an elementary matrix is elementary:
$E_{ij}^T = E_{ij}$, $E_{ij}(m)^T = E_{ji}(m)$, $E_i(m)^T = E_i(m)$
A matrix A is **symmetric** if $A^T = A$
An occasionally useful fact: AE, where E is an elementary matrix, is the result of an elementary **column** operation on A .

The transpose and rank:
For any pair of compatible matrices, r(AB) $\leq$ r(A)
Consequences: r(A^T) = r(A) for any matrix A; r(AB) $\leq$ r(B), as well.

Matrix inverses

One way to solve Ax=b : find a matrix B with BA=I . When is there such a matrix?
(Think about square matrices...) A an n-by-n matrix ; n=r(I)=r(BA)$\leq$r(A)$\leq$n implies that r(A)=n . This is necessary, and it is also sufficient!
r(A)=n, then the RREF of A has n pivots in n rows and columns, so has a pivot in every row, so the RREF of A is I. But! this means we can get to I from A by row operations, which correspond to multiplication by elementary matrices. *So* multiply A (on the left) by the correct elementary matrices and you get I; call the product of those matrices B and you get BA=I !
A matrix B is an **inverse** of A if AB=I and BA=I; it turns out, the inverse of a matrix is always unique. We call it A^{-1} (and call A invertible).
Finding A^{-1} : row reduction! (of course...)
Build the "super-augmented" matrix (A|I) (the matrix A with the identity matrix next to it). Row reduce A, and carry out the operations on the entire row of the S-A matrix (i.e., carry out the identical row operations on I). Wnem done, if invertible+ the left-hand side of the S-A matrix will be I; the right-hand side will be A^{-1} !
I.e., if (A|I) $\rightarrow$ (I|B) by row operations, then I=BA .

Basic facts:
$(A^{-1})^{-1} = A$
if A and B are invertible, then so is AB, and $(AB)^{-1}$ - $B^{-1}A^{-1}$
$(cA)^{-1} = (1/c)A^{-1}$
$(A^T)^{-1} = (A^{-1})^T$
If A is invertible, and $AB = AC$, then $B = C$; if $BA = CA$, then $B = C$.
Inverses of elementary matrices:
$E_{ij}^{-1} = E_{ij}$, $E_{ij}(m)^{-1} = E_{ij}(-m)$, $E_i(m)^{-1} = E_i(1/m)$
Highly useful formula: for a 2-by-2 matrix,

A= $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ and D=ad-bc , $A^{-1} = \frac{1}{D}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$

(Note: need D=ad-bc $\neq$ 0 for this to work....)
Some conditions for/consequences of invertibility: the following are all equivalent (A = n-by-n matrix).

1. A is invertible,
2. r(A) = n.
3. The RREF of A is I_n.
4. Every linear system Ax=b has a unique solution.
5. For one choice of b, Ax=b has a unique solution (i.e., if one does, they all do...).
6. The equation Ax=0 has only the solution x=0.
7. There is a matrix B with BA=I.

The equivalence of 4. and 6. is sometimes stated as **Fredholm's alternative**: Either every equation Ax=b has a unique solution, or the equation Ax=0 has a **non-trivial** solution (and only one of the alternatives can occur).

Subspaces, bases, dimension, and rank

Basic idea: $W \subseteq \mathbb{R}^n$ is a subspace if whenever $c \in \mathbb{R}$ and $u, v \in W$, we **always** have $cu, u+v \in W$ (W is "closed" under addition and scalar multiplication).

Examples: $\{(x,y,z) \in \mathbb{R}^3 : z=0\}$ is a subspace of $\mathbb{R}^3$
$\{(x,y,z) \in \mathbb{R}^3 : z=1\}$ is **not** a subspace of $\mathbb{R}^3$

Basic construction: $v_1, \cdots, v_n \in V$
$W = \{a_1v_1 + \cdots a_nv_n : a_1, \ldots, a_n \in \mathbb{R}$ = all linear combinations of $v_1, \cdots, v_n = \text{span}\{v_1, \cdots, v_n\}$
= the *span* of $v_1, \cdots, v_n$, is a subspace of $\mathbb{R}^k$

Basic fact: if $w_1, \ldots, w_k \in \text{span}\{v_1, \cdots, v_n\}$, then $\text{span}\{w_1, \cdots, w_k\} \subseteq \text{span}\{v_1, \cdots, v_n\}$

Subspaces from matrices
column space of $A = \text{col}(A) = \text{span}\{\text{the columns of } A\}$
row space of $A = \text{row}(A) = \text{span}\{(\text{transposes of the }) \text{ rows of } A\}$
nullspace of $A = \text{null}(A) = \{x \in \mathbb{R}^n : Ax = 0\}$

(Check: null(A) is a subspace!)

Alternative view Ax = lin comb of columns of A, so is in col(A); in fact, $\text{col}(A) = \{Ax : x \in \mathbb{R}^n\}$. So col($A$) is the set of vectors b for which $Ax = b$ has a solution. Any two solutions $Ax = b = Ay$ have $A(x-y) = AX - Ay = b - b = 0$, so $x - y$ is in null(A). So the collection of <u>all</u> solutions to $AX = b$ are (particular solution)+(vector in null(A)). So col(A) tells is which SLEs have solutions, and null(A) tells us how many solutions there are.

Bases:

A **basis** for a subspace V is a set of vectors $v_1, \ldots, v_n$ so that (a) they are linearly independent, and (b) V=span$\{v_1, \ldots, v_n\}$.
The idea: a basis allows you to express every vector in the subspace as a linear combination <u>in exactly</u> <u>one</u> <u>way</u>.

A system of equations $Ax = b$ has a solution iff $b \in$col(A) .

If $Ax_0 = b$, then every other solution to $Ax = b$ is $x = x_0 + z$, where $z \in$null(A) .

The row, column, and nullspaces of a matrix A are therefore useful spaces (they tell us useful things about solutions to the corresponding linear system), so it is useful to have bases for them.

Finding a basis for the row space.
Basic idea: if B is obtained from A by elementary row operations, then row(A) =row(B).
So of R is the reduced row echelon form of A, row(R) =row(A)
But a basis for row(R) is quick to identify; take all of the non-zero rows of R ! (The zero rows are clearly redundant.) These rows are linearly independent, since each has a 'special coordinate' where, among the rows, only it is non-zero. That coordinate is the *pivot* in that row. So in any linear combination of rows, only that vector can contribute something non-zero to that coordinate. *Consequently*, in any linear combination, that coordinate is the **coefficient** of our vector! **So**, if the lin comb is $\vec{0}$, the coefficient of our vector (i.e., each vector!) is 0.

Put bluntly, to find a basis for row(A), row reduce A, to R; the (transposes of) the non-zero rows of R form a basis for row(A).

This in turn gives a way to find a basis for col(A), since col(A) =row(A^T) !

To find a basis for col(A), take A^T, row reduce it to S; the (transposes of) the non-zero rows of S form a basis for row(A^T) =col(A) .

This is probably in fact the most useful basis for col(A), since each basis vector has that special coordinate. This makes it very quick to decide if, for any given vector b, $Ax = b$ has a solution. You need to decide if b can be written as a linear combination of your basis vectors; but each coefficient will be the coordinate of b lying at the special coordinate of each vector. Then just check to see if **that** linear combination of your basis vectors adds up to b !

There is another, perhaps less useful, but faster way to build a basis for col(A); row reduce A to R, locate the pivots in R, and take the columns of A (Note: A, **not** R !) that correspond to the columns containing the pivots. These form a (different) basis for col(A).

Why? Imagine building a matrix B out of just the pivot columns. Then in row reduced form there is a pivot in every column. Solving $Bv = \vec{0}$ in the case that there are no free variables, we get $v = \vec{0}$, so the columns are linearly independent. If we now add a free column to B to get C, we get the same collection of pivots, so our added column represents a free variable. Then there are non-trivial solutions to $Cv = \vec{0}$, so the columns of C are not linearly independent. This means that the added columns can be expressed as a linear combination of the bound columns. This is true for all free columns, so the bound columns span col(A).

Finally, there is the nullspace null(A). To find a basis for null(A):

Row reduce A to R, and use each row of R to solve $Rx = \vec{0}$ by expressing each bound variable in terms of the frees. collect the coefficients together and write $x = x_{i_1}v_1 + \cdots + x_{i_k}v_k$ where the x_{i_j} are the free variables. Then the vectors $v_1, \ldots , v_k$ form a basis for null(A).

Why? By construction they span null(A); and just as with our row space procedure, each has a special coordinate where only it is 0 (the coordinate corresponding to the free variable!).

Note: since the number of vectors in the bases for row(A) and col(A) is the same as the number of pivots (= number of nonzero rows in the RREF) = rank of A, we have dim(row(A))=dim(col(A))=$r(A)$. And since the number of vectors in the basis for null(A) is the same as the number of free variables for A (= the number of columns without a pivot) = nullity of A (hence the name!), we have dim(null(A)) = $n(A) = n - r(A)$ (where n=number of columns of A).
So, dim(col(A)) + dim(null(A)) = the number of columns of A .