CS205 – Class 2

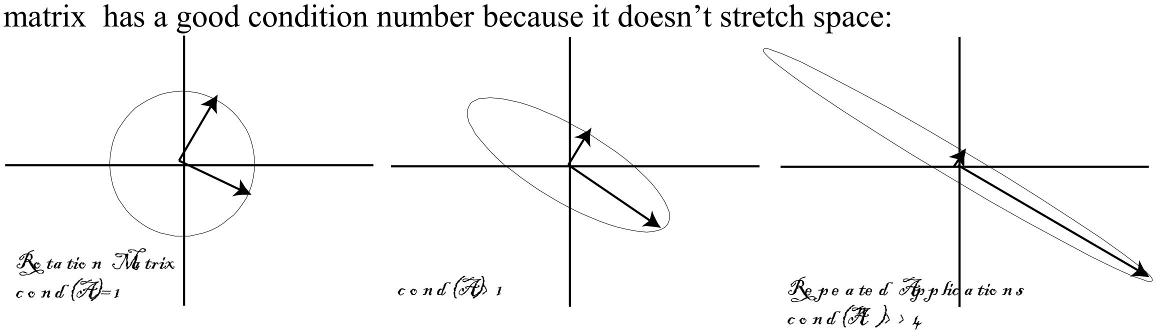
Linear Systems Continued

Covered in class: all sections

1. When constructing M_k we needed to divide by a_k which is the element on the diagonal. This could pose difficulties if the diagonal element was zero. For example, consider the matrix equation $\begin{bmatrix} 0 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 2 \\ 8 \\ 10 \end{bmatrix}$ where forming M_1 would lead to a division by $a_{11} = 0$. In general, LU factorization fails if a small number shows up on the diagonal at any stage.
    a. When small numbers occur on the diagonal, one can change rows. In general, one can switch the current row k with a row j below it with j > k in order to get a larger diagonal element. This is called **pivoting. Partial pivoting** is the process of switching rows to *always* get the largest diagonal element, and **full pivoting** consists of switching both rows and columns to always obtain the largest possible diagonal element. Note that when switching columns, the order of the elements in the unknown vector needs to be changed in the obvious way, i.e. there is a corresponding row switching in the vector of unknowns.
    b. A **permutation matrix** can be used to switch rows in a matrix. Permutation matrices can be constructed by performing the desired row switch on the identity matrix. For example, a permutation matrix that switches the first and third rows of a matrix is obtained by switching the first and third rows of the identity matrix. For a 3x3 matrix, $P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$.
        i. $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix} = \begin{bmatrix} -2 & -3 & 7 \\ 4 & 9 & -3 \\ 0 & 4 & 2 \end{bmatrix}$
        ii. Switching two rows twice puts the rows back, so P is its own inverse (this generalizes to arbitrary permutation matrices as $P^{-1} = P^T$ since P is an orthogonal matrix; in this simple 2-row flip case the matrix P happens to be symmetric, therefore $P^{-1} = P^T = P$)
        iii. Partial pivoting gives A=LU with $U = M_{n-1}P_{n-1} \cdots M_1 P_1 A$ and $L = P_1 L_1 \cdots P_{n-1} L_{n-1}$ where U is upper triangular, but L is a *permutation* of lower triangular matrix.
        iv. It turns out that that we can rewrite L as $L = P_1 \cdots P_{n-1} L_1^P \cdots L_{n-1}^P$ where each $L_k^P = I + (P_{n-1} \cdots P_{k+1} m_k) e_k^T$ has the same form as L_k. Thus, we can write $PA = L^P U$ where $L^P = L_1^P \cdots L_{n-1}^P$ *is lower triangular* and $P = P_{n-1} \cdots P_1$ is the total permutation matrix.
        v. Thus, we could figure out all the permutations and do those first rewriting Ax=b as PAx=Pb and then factorizing into LUx=Pb to solve for x.
2. Certain types of matrices do not require pivoting, e.g. *symmetric positive definite* and *diagonally dominant* matrices.
    a. A matrix is symmetric if $A=A^T$

b. A matrix is positive definite if if $x^T Ax > 0$ for all $x \neq 0$
c. A matrix is **diagonally dominant** if the magnitude of the diagonal element is strictly larger than the sum of the magnitudes of all the other elements in its row, and strictly larger than the sum of the magnitudes of all the other elements its column.

3. The inverse of a matrix can be found by solving AX=I with X= A^{-1}. In particular, we can first find the permutation matrix and write PAX=PI=P, and then obtain the LU factorization to obtain LUX=P. Finally, one simply solves $LUx_k = p_k$ separately for each column of X.
4. The most commonly used vector norms are $\|x\|_1 = \sum_i |x_i|$, $\|x\|_2 = \sqrt{\sum_i x_i^2}$, and $\|x\|_\infty = \max_i |x_i|$.
5. A vector norm induces a corresponding matrix norm in accordance with the definition: $\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}$
6. The corresponding matrix norms are as follows:
   a. $\|A\|_1 = \max_j \sum_i |a_{ij}|$ which is the maximum absolute column sum.
   b. $\|A\|_\infty = \max_i \sum_j |a_{ij}|$ which is the maximum absolute row sum.
   c. $\|A\|_2$ is the square root of the maximum eigenvalue of $A^T A$ (i.e., the largest singular value of A).
   d. In some sense all norms are "equivalent". That is, they are *interchangeable* for many theoretical pursuits.
   e. In another sense, however, all norms are not equivalent. It is important to be aware of what a specific norm measures. All too often a misapplied norm will be used to legitimize undesirable results. For example, minimizing an L_2 norm for the nodes one a piece of simulated cloth could yield wildly unsatisfactory results. One node on the moon and the rest on earth should NOT be considered a successful simulation.
7. The **condition number** for solving the problem Ax=b for the matrix A is $\|A\| \, \|A^{-1}\|$ for nonsingular matrices, or ∞ for singular matrices. Note that it doesn't matter what b is.
   a. The condition number is always greater than or equal to 1
   b. The condition number of the identity is 1
   c. The condition number of a singular matrix is infinity
8. View a matrix as warping space toward a dominant eigenvector direction Repeated applications approach a projection onto the line of that eigenvector. When solving Ax=b your b corresponds to a point in that warped space. As that b gets compressed more toward the line it becomes harder to find the exact x that will correspond because a small perturbation in x leads to something very close in b. Thus a orthogonal (rotation) matrix has a good condition number because it doesn't stretch space:



9. If a matrix is both symmetric and positive definite (which happens quite bit), we can obtain a **Cholesky factorization** of A= LL^T

a. Suppose that we write down $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} \\ 0 & l_{22} \end{bmatrix}$ in order to represent A=LL^T for a 2x2 matrix. Then by multiplying out the right hand side we obtain $\begin{bmatrix} a_{11} & a_{21} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} (l_{11})^2 & l_{11}l_{21} \\ l_{11}l_{21} & (l_{21})^2 + (l_{22})^2 \end{bmatrix}$. Thus we can solve for $l_{11} = \sqrt{a_{11}}$, $l_{21} = a_{21}/l_{11}$, and $l_{22} = \sqrt{a_{22} - (l_{21})^2}$.

b. Then general algorithm is as follows:
   i. for(j=1,n) {
   ii. for(k=1,j-1) for(i=j,n) $a_{ij} -= a_{ik}a_{jk}$;
   iii. $a_{jj} = \sqrt{a_{jj}}$; for(k=j+1,n) $a_{kj} /= a_{jj}$ }

c. In other words it's the same as
   i. For each column j of the matrix.
   ii. Loop over all previous columns k, and subtract a multiple of column k from the current column j.
   iii. Take the square root of the diagonal entry, and scale column j by that value.

d. Note that this algorithm above factors the matrix "in place".

e. In the first class we discussed different types of errors (e.g., roundoff, truncation, modeling, etc.). It is very important to be aware of what leading sources of error are when using numerical algorithms. For example, in a seminal cloth simulation paper for computer graphics by Baraff and Witkin they encountered a matrix equation, Ax=b, which was difficult to solve. They simply discarded the non-symmetric part of A and symmetrized it reducing their problem to one which can use more effective techniques like a Cholesky decomposition.

f. One way of symmetrizing a matrix A is taking $(A+A^T)/2$. Notice that this does not change the diagonal entries, and simply averages the off diagonal entries.