COMPUTER SCIENCE 349A
Handout Number 17

MATRIX INVERSES

This topic is discussed in the textbook in Section 10.2 in terms of an LU decomposition (which is just another way of interpreting Gaussian elimination). We are omitting all of Chapter 10. The following material is similar to that in Section 10.2 but is not described in terms of an LU decomposition.

If it is necessary to compute A^{-1}, this is most efficiently and accurately done **using Gaussian elimination (with partial pivoting)** and the fact that $A\,A^{-1} = I$.

Suppose that a matrix A is given. If the (unknown) column vectors of A^{-1} are denoted $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$ then

$$A\,A^{-1} = A \times \left[x^{(1)} \mid x^{(2)} \mid \cdots \mid x^{(n)}\right] = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix},$$

and the vectors $x^{(i)}$ can be determined by solving the n linear systems

$$Ax^{(1)} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix},\ Ax^{(2)} = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix},\ \cdots,\ Ax^{(n)} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}.$$

These n linear systems can be efficiently solved since they all have the same coefficient matrix A.

Algorithm for computing A^{-1}: apply Gaussian elimination (with partial pivoting) to the $n \times 2n$ augmented matrix $[A \mid I\,]$. This corresponds to having n right-hand side vectors, and will require that n back substitutions be done. However, the forward elimination (the reduction of A to upper triangular form) needs to be done only once.

Cost (in terms of a floating-point operation count) is approximately $\frac{4}{3}n^3$ multiplications/divides and the same number of additions/subtractions. However, both of these operation counts can be reduced to about n^3 by avoiding multiplications by 1 and 0, and additions to 0. Thus, the **total flop count** is approximately $2n^3$ flops.

Important comment. In general you should avoid computation of a matrix inverse if, as is often the case, possible.

Examples.

1. If you are given a nonsingular matrix A and a vector b, and you need to compute $A^{-1}b$, do not compute A^{-1} and multiply this by b. Instead, you should solve the linear system $Ax = b$ for x using Gaussian elimination (with partial pivoting). Note that $x = A^{-1}b$.

Consider the relative costs of these two methods of solution.

(i) Compute A^{-1} (requires about $2n^3$ flops)
Compute $A^{-1}b$ (requires about $2n^2$ flops)
Total is $2n^3 + O(n^2)$ flops.

(ii) Solve $Ax = b$ for x using Gaussian elimination. Requires only $\frac{2n^3}{3} + O(n^2)$ flops.

2. If you are given a nonsingular matrix A and a matrix B, and you need to compute $A^{-1}B$, for the same reasons as above you should not compute A^{-1} and multiply it times B. It is much more efficient to solve the linear system $AX = B$ for the matrix X; that is, apply Gaussian elimination (with partial pivoting) to the augmented matrix $[A \mid B]$.

Note. A matrix inverse should be computed only if the individual entries of A^{-1} have significance in an application. There is some discussion of this in the context of an Engineering application on pages 286-287 of the 6th ed. or pages 290-291 of the 7th ed., and this is true in some statistical applications. But if a matrix inverse is used only in some product of A^{-1} with another matrix or vector, then there is no need to compute the inverse.