Mathematical Methods (10/24.539)

III. Overview of Linear Algebra

Introduction

The subject of ***Linear Algebra***, in general, covers a broad range of topics. Our goal in this unit is simply to review the standard concepts needed for other subjects in this course. In particular, we will first cover the basic notation and operations associated with vector and matrix algebra and then focus on systems of linear equations, including both inhomogeneous and homogeneous equations. The homogeneous systems, of course, also lead to the important subject of eigenvalues and eigenvectors of a matrix. The classification of matrices based on the properties of their eigenvalues and eigenvectors is also discussed briefly. Finally, a demo is provided to illustrate how to perform many of the analytical operations outlined in this unit within the Matlab programming language.

The subjects reviewed here are treated as part of most undergraduate engineering programs (although you may not have had a formal course in Linear Algebra). As such, this section of notes is primarily intended as review material. Students already comfortable with this subject material should quickly browse this section to become familiar with the specific notation used here, and also to become acquainted with performing many common and very useful matrix-vector operations within the Matlab environment. Students weak in this subject are encouraged to study these notes thoroughly and to consult other reference books on this subject as needed.

The key topics treated here are:

Note: The general subject area of linear algebra is introduced as part of my undergraduate ***Applied Problem Solving with Matlab*** course. As part of that course, I have provided a number of Matlab demos and illustrative applications on my course website. If you feel you need some additional study in this area, you might check out www.profjrwhite.com/courses.htm to see if the material there is helpful???

Basic Notation and Operations

This section introduces some terminology and notation from linear algebra and also outlines some basic arithmetic operations with vectors and matrices. Let's start by defining some notation associated with vectors and matrices.

Note: In the technical literature, vectors and matrices are usually written with **bold** lower and upper case letters or as variables with a single underline or double underline, respectively, for 1-D vectors or 2-D matrices. In my courses, I use both of these notation schemes so that the students become familiar with a variety of forms for representing these quantities. In this section, however, I have tried to use the underline notation consistently so that it is not too confusing for the student who has not used their linear algebra background for some time…

A ***vector*** is simply an ordered set of numbers or quantities. A column vector is usually written as

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

and a row vector is given by

$$\underline{x}^T = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}$$

where the length of the vector is equal to the number of elements. The usual notation, without the superscript T, refers to the multiple row, single column format - thus, the vector quantity is referred to as a column vector. Similarly, the row vector has only one row but multiple columns.

Given two column vectors, $\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ and $\underline{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$, the most common arithmetic operations are defined as follows:

Addition

$$\underline{z} = \underline{x} + \underline{y} = \begin{bmatrix} x_1 + y_1 \\ x_2 + y_2 \\ x_3 + y_3 \end{bmatrix} \qquad \text{or} \qquad z_i = x_i + y_i \tag{3.1}$$

Multiplication by a Scalar

$$\underline{z} = \alpha \underline{x} = \begin{bmatrix} \alpha x_1 \\ \alpha x_2 \\ \alpha x_3 \end{bmatrix} \qquad \text{or} \qquad z_i = \alpha x_i \tag{3.2}$$

Dot Product (inner product)

$$\alpha = \underline{x} \cdot \underline{y} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = x_1 y_1 + x_2 y_2 + x_3 y_3 \qquad \text{or} \qquad \alpha = \underline{x}^T \underline{y} = \sum_i x_i y_i \tag{3.3}$$

Outer Product

$$\underline{\underline{A}} = \underline{x}\,\underline{y}^T = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} = \begin{bmatrix} x_1y_1 & x_1y_2 & x_1y_3 \\ x_2y_1 & x_2y_2 & x_2y_3 \\ x_3y_1 & x_3y_2 & x_3y_3 \end{bmatrix}$$

or $\quad \underline{\underline{A}} = \left[a_{ij}\right] \quad$ where $\quad a_{ij} = x_i y_j \qquad$ (3.4)

A ***matrix*** is a regular 2-D array of numbers or quantities and is denoted with a double underline,

$$\underline{\underline{A}} = \left[a_{ij}\right], \quad \underline{\underline{B}} = \left[b_{ij}\right], \quad \text{etc.}$$

where i is the row index and j is the column index. For example, a 3x3 matrix can be written as

$$\underline{\underline{A}} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Again, many of the common arithmetic operations with matrices are as follows:

Addition

$$\underline{\underline{C}} = \underline{\underline{A}} + \underline{\underline{B}} \qquad \text{or} \qquad c_{ij} = a_{ij} + b_{ij} \tag{3.5}$$

Scalar Multiplication

$$\underline{\underline{C}} = \alpha\underline{\underline{A}} \qquad \text{or} \qquad c_{ij} = \alpha a_{ij} \tag{3.6}$$

Matrix Multiplication

$$\underline{\underline{C}} = \underline{\underline{A}}\,\underline{\underline{B}} \qquad \text{or} \qquad c_{ij} = \sum_k a_{ik} b_{kj} \tag{3.7}$$

where the number of columns of the first matrix must be equal to the number of rows of the second matrix, or

$$\begin{matrix} \underline{\underline{A}} & \times & \underline{\underline{B}} & = & \underline{\underline{C}} \\ (m \times n) & & (n \times p) & = & (m \times p) \end{matrix}$$

where the notation $m \times n$, for example, implies that the matrix has m rows and n columns.

Matrix-Vector Multiplication

$$\underline{y} = \underline{\underline{A}}\,\underline{x} \qquad \text{or} \qquad y_i = \sum_j a_{ij} x_j \tag{3.8}$$

Matrix Transpose

$$\underline{\underline{C}} = \underline{\underline{A}}^T \qquad \text{or} \qquad c_{ij} = a_{ji} \tag{3.9}$$

Also there are a number of special matrices of interest. For example, some of these matrices include diagonal, triangular, square, and identity matrices, as well as symmetric and skew symmetric matrices, etc.. Most of the names for these matrices are self-explanatory. A square matrix is one with an equal number of rows and columns. A lower triangular matrix is a square matrix with all zero elements above the diagonal elements. Also, a real symmetric matrix is one that satisfies

$$\underline{\underline{A}}^T = \underline{\underline{A}} \qquad \text{or} \qquad a_{ji} = a_{ij} \tag{3.10}$$

and a real skew-symmetric matrix satisfies the relationship

$$\underline{\underline{A}}^T = -\underline{\underline{A}} \qquad \text{or} \qquad a_{ji} = -a_{ij} \tag{3.11}$$

Other relationships will be defined as needed in subsequent subsections.

Systems of Linear Algebraic Equations

General Notation

The major motivation for the matrix/vector notation outlined in the previous section is as a shorthand representation for linear systems of algebraic equations. The system of equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1N}x_N &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2N}x_N &= b_2 \\ &\vdots \\ a_{N1}x_1 + a_{N2}x_2 + \cdots + a_{NN}x_N &= b_N \end{aligned} \tag{3.12}$$

can be written in matrix notation as

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ & \vdots & \vdots & \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} \tag{3.13}$$

and, using the definitions of matrix multiplication, we have

$$\underline{\underline{A}}\,\underline{x} = \underline{b} \tag{3.14}$$

Gauss Elimination

Most direct methods for solving systems of equations involve a sequence of ***elementary row operations***. These operations represent legal algebraic manipulations that do not alter the basic equality associated with the original equations. The purpose of the row operations is to take the original equations and put them into a form that is easier to solve than the original equations. There are three row operations that are used to systematically simplify the original system of equations:

1. Interchange two rows.
2. Multiply a row by a constant.
3. Add a constant times one row to another row.

The ***Gauss Elimination Method*** is the most well known method that implements these row operations in a systematic manner to take the original system and convert the matrix to upper triangular form. In this form, back substitution is used to evaluate the unknown solution vector $\underline{x}$. The elimination step can be represented symbolically using an augmented matrix notation, $\underline{\underline{\tilde{A}}} = \left[\underline{\underline{A}} \quad \underline{b}\right]$, or, for example, a 3x3 system would be transformed as follows:

$$\begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} \Rightarrow \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \end{bmatrix}$$

where the x notation implies a general entry and the last column in the original matrix contains the right hand side vector $\underline{b}$. Of course, after transformation, the entries in the resultant matrix are different from the original case. However, this upper triangular form (for the n×n part of the augmented matrix), which is also known as ***echelon form***, is an equivalent representation of the original equation. Once in this form, one can easily use back substitution to solve for the unknown vector $\underline{x}$.

As a simple example of this method, consider the following 3x3 system:

$$\underline{\underline{A}} \quad \underline{x} = \underline{b}$$

$$\begin{bmatrix} 2 & 1 & 9 \\ -2 & 3 & -1 \\ 4 & 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}$$

Written in augmented matrix form, this becomes

$$\begin{bmatrix} 2 & 1 & 9 & 1 \\ -2 & 3 & -1 & 2 \\ 4 & 2 & 1 & 0 \end{bmatrix}$$

Now, as our first row operation, take row 1 added to row 2 to give

$$\begin{bmatrix} 2 & 1 & 9 & 1 \\ 0 & 4 & 8 & 3 \\ 4 & 2 & 1 & 0 \end{bmatrix}$$

Note that only row 2 is modified in this step. Now take -2 times row 1 added to row 3 to give

$$\begin{bmatrix} 2 & 1 & 9 & 1 \\ 0 & 4 & 8 & 3 \\ 0 & 0 & -17 & -2 \end{bmatrix}$$

This system is now in echelon form. Using back substitution gives

$$x_3 = \frac{-2}{-17} = \frac{2}{17}$$

$$x_2 = \frac{[3 - 8x_3]}{4} = \frac{35}{68}$$

$$x_1 = \frac{[1 - x_2 - 9x_3]}{2} = -\frac{39}{136}$$

A quick check shows that this is indeed the correct solution to the original system of equations. The reader is referred to the section on Numerical Solution of Algebraic Equations for a more detailed treatment of the Gauss Elimination Algorithm and other techniques for solving systems of linear algebraic equations on the computer.

Determinant of a Matrix

The determinant of a matrix, denoted as $\det \underline{\underline{A}}$ or $|\underline{\underline{A}}|$, appears frequently in applications of matrix equations. It is sometimes thought of as a ***measure of the size or magnitude*** of a matrix. Independent of its formal interpretation, it does appear in many formal definitions of other quantities and we must be able to compute $\det \underline{\underline{A}}$ in lots of situations. For hand manipulation of low order systems, Laplace's expansion for $\det \underline{\underline{A}}$ is the best way to evaluate this quantity (computer computation is done differently and more efficiently by other means).

Laplace's expansion can be written in terms of an expansion along any row i as

$$\det \underline{\underline{A}} = \sum_j a_{ij} c_{ij} \qquad \text{for any i} \tag{3.15}$$

or down any column j as

$$\det \underline{\underline{A}} = \sum_i a_{ij} c_{ij} \qquad \text{for any j} \tag{3.16}$$

where c_{ij} is the ***cofactor*** of element a_{ij}. The elements of the cofactor matrix are defined as

$$c_{ij} = (-1)^{i+j} M_{ij} \tag{3.17}$$

where M_{ij} is referred to as the ***minor*** of the a_{ij} element. M_{ij} is defined as the determinant of the matrix formed by deleting the i^{th} row and the j^{th} column from the original matrix.

For example, given a general 3x3 matrix

$$\underline{\underline{A}} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

we can expand down column 1 (for example), giving

$$\det \underline{\underline{A}} = a_{11}c_{11} + a_{21}c_{21} + a_{31}c_{31}$$

where

$$c_{11} = (+1) M_{11} = \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} = a_{22}a_{33} - a_{23}a_{32}$$

$$c_{21} = (-1) M_{21} = -\begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} = -a_{12}a_{33} + a_{13}a_{32}$$

$$c_{31} = (+1) M_{31} = \begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix} = a_{12}a_{23} - a_{13}a_{22}$$

Note that this is exactly the same result as if one expands along row 1 (or any other row or column).

The determinant of a matrix may or may not be altered under certain variations to the original matrix. Several important relations are as follows:

1. The $\det \underline{\underline{A}}$ is not altered if the rows are written as columns in the same order. Therefore,

$$\det \underline{\underline{A}} = \det \underline{\underline{A}}^T \tag{3.18}$$

2. If any two rows or columns are interchanged, the value of $\det \underline{\underline{A}}$ is multiplied by -1.

3. The value of $\det \underline{\underline{A}}$ is not altered if the elements of one row are altered by adding any constant multiple of another row to them.

4. The determinant is multiplied by a constant α if any row is multiplied by α.

5. The determinant of a diagonal matrix is simply the product of the diagonal elements. This is also true for triangular matrices.

6. For square matrices,

$$\det\left(\underline{\underline{A}}\,\underline{\underline{B}}\right) = \det\left(\underline{\underline{B}}\,\underline{\underline{A}}\right) = \det \underline{\underline{A}} \det \underline{\underline{B}} \tag{3.19}$$

Matrix Inverse

The matrix inverse, denoted as $\underline{\underline{A}}^{-1}$, is a quantity used in the formal manipulation and solution of systems of equations. $\underline{\underline{A}}^{-1}$ is defined such that $\underline{\underline{A}}^{-1}\underline{\underline{A}} = \underline{\underline{A}}\,\underline{\underline{A}}^{-1} = \underline{\underline{I}}$. This says that a square matrix multiplied by its inverse gives the identity matrix. Also, the identity matrix operating on a matrix or vector of appropriate size does not alter the original quantity. These facts can be used to write the formal solution to a system of equations. In particular, given $\underline{\underline{A}}\,\underline{x} = \underline{b}$, a formal solution for $\underline{x}$ can be developed as follows:

Starting with $\underline{\underline{A}}\,\underline{x} = \underline{b}$, pre-multiply both sides by $\underline{\underline{A}}^{-1}$ to give

$$\underline{\underline{A}}^{-1}\underline{\underline{A}}\,\underline{x} = \underline{\underline{A}}^{-1}\underline{b}$$

but $\underline{\underline{A}}^{-1}\underline{\underline{A}} = \underline{\underline{I}}$, and $\underline{\underline{I}}\,\underline{x} = \underline{x}$, therefore we have

$$\underline{x} = \underline{\underline{A}}^{-1}\underline{b} \tag{3.20}$$

This formal solution is very important, since it provides a basis for discussing the uniqueness and existence of solutions and it also allows for various manipulations of matrix equations. However, the reader should be cautioned that this formulation is not the most efficient procedure for actually computing the solution vector $\underline{x}$. For computer implementation, especially for large systems, other techniques are far more efficient (see Section VI on Numerical Solution of Algebraic Equations).

There are many cases, however, when it is useful to actually evaluate the inverse matrix. There are a variety of ways to do this. For low order systems, the following formula is often applied,

$$\underline{\underline{A}}^{-1} = \frac{\underline{\underline{C}}^T}{\det \underline{\underline{A}}} \tag{3.21}$$

where $\underline{\underline{C}}$ is the matrix whose elements are the cofactors of $\underline{\underline{A}}$. For larger systems and for automated implementation on the computer, some form of the Gauss-Jordan Method is often used. The Gauss-Jordan Elimination Method (which is just an extension of the Gauss Elimination technique discussed earlier) applies elementary row operations to transform the original augmented matrix as follows:

$$\begin{bmatrix} x & x & x & 1 & 0 & 0 \\ x & x & x & 0 & 1 & 0 \\ x & x & x & 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 0 & x & x & x \\ 0 & 1 & 0 & x & x & x \\ 0 & 0 & 1 & x & x & x \end{bmatrix}$$

With this symbolic notation, we see that the original matrix is augmented with the identity matrix. Extending the notation from before, this says we are trying to evaluate a matrix equation of the form, $\underline{\underline{A}}\,\underline{\underline{X}} = \underline{\underline{I}}$, for the unknown matrix $\underline{\underline{X}}$. Therefore, we know, from the definition of the inverse matrix, that $\underline{\underline{X}} = \underline{\underline{A}}^{-1}$. We can solve for $\underline{\underline{X}}$ by performing row operations on the augmented matrix $\left[\underline{\underline{A}} \quad \underline{\underline{I}}\right]$, finally putting it into the form $\left[\underline{\underline{I}} \quad \underline{\underline{X}}\right]$.

Let's illustrate these two methods for finding the inverse matrix using the following 3x3 matrix,

$$A = \begin{bmatrix} 3 & -2 & 2 \\ 1 & 2 & -3 \\ 4 & 1 & 2 \end{bmatrix}$$

Method I:

Using eqn. (3.21) let's first find $\det \underline{\underline{A}}$ by expanding along row 1, or

$$\det \underline{\underline{A}} = 3\begin{vmatrix} 2 & -3 \\ 1 & 2 \end{vmatrix} + 2\begin{vmatrix} 1 & -3 \\ 4 & 2 \end{vmatrix} + 2\begin{vmatrix} 1 & 2 \\ 4 & 1 \end{vmatrix} = 3(7) + 2(14) + 2(-7) = 35$$

Also the cofactor matrix is given by

$$\underline{\underline{C}} = \begin{bmatrix} 7 & -14 & -7 \\ 6 & -2 & -11 \\ 2 & 11 & 8 \end{bmatrix}$$

Therefore,

$$\underline{\underline{A}}^{-1} = \frac{\underline{\underline{C}}^{T}}{\det \underline{\underline{A}}} = \frac{1}{35}\begin{bmatrix} 7 & 6 & 2 \\ -14 & -2 & 11 \\ -7 & -11 & 8 \end{bmatrix}$$

and a quick check on $\underline{\underline{A}}^{-1}\underline{\underline{A}} \overset{?}{=} \underline{\underline{I}}$ gives

$$\frac{1}{35}\begin{bmatrix} 7 & 6 & 2 \\ -14 & -2 & 11 \\ -7 & -11 & 8 \end{bmatrix}\begin{bmatrix} 3 & -2 & 2 \\ 1 & 2 & -3 \\ 4 & 1 & 2 \end{bmatrix} = \frac{1}{35}\begin{bmatrix} 35 & 0 & 0 \\ 0 & 35 & 0 \\ 0 & 0 & 35 \end{bmatrix} = \underline{\underline{I}}$$

which shows that all the manipulations have been done correctly!

Method II:

For the Gauss-Jordan method, we start with the original matrix augmented with the 3x3 identity matrix, or

$$\begin{bmatrix} 3 & -2 & 2 & 1 & 0 & 0 \\ 1 & 2 & -3 & 0 & 1 & 0 \\ 4 & 1 & 2 & 0 & 0 & 1 \end{bmatrix}$$

Now let's systematically perform a set of row operations to transform this matrix into the desired form. To start, take -4/3 times row 1 added to row 3, giving

$$\begin{bmatrix} 3 & -2 & 2 & 1 & 0 & 0 \\ 1 & 2 & -3 & 0 & 1 & 0 \\ 0 & 11/3 & -2/3 & -4/3 & 0 & 1 \end{bmatrix}$$

Now take -1/3 times row 1 added to row 2, to give

$$\begin{bmatrix} 3 & -2 & 2 & 1 & 0 & 0 \\ 0 & 8/3 & -11/3 & -1/3 & 1 & 0 \\ 0 & 11/3 & -2/3 & -4/3 & 0 & 1 \end{bmatrix}$$

Taking -11/8 times row 2 added to row 3 gives

$$\begin{bmatrix} 3 & -2 & 2 & 1 & 0 & 0 \\ 0 & 8/3 & -11/3 & -1/3 & 1 & 0 \\ 0 & 0 & 35/8 & -7/8 & -11/8 & 1 \end{bmatrix}$$

Now if each row is normalized, we have

$$\begin{bmatrix} 1 & -2/3 & 2/3 & 1/3 & 0 & 0 \\ 0 & 1 & -11/8 & -1/8 & 3/8 & 0 \\ 0 & 0 & 1 & -7/35 & -11/35 & 8/35 \end{bmatrix}$$

Continuing to perform row operations to eliminate the upper triangular terms, we take -2/3 times row 3 added to row 1 to give

$$\begin{bmatrix} 1 & -2/3 & 0 & 49/105 & 22/105 & -16/105 \\ 0 & 1 & -11/8 & -1/8 & 3/8 & 0 \\ 0 & 0 & 1 & -7/35 & -11/35 & 8/35 \end{bmatrix}$$

Then 11/8 times row 3 added to row 2 gives

$$\begin{bmatrix} 1 & -2/3 & 0 & 49/105 & 22/105 & -16/105 \\ 0 & 1 & 0 & -14/35 & -2/35 & 11/35 \\ 0 & 0 & 1 & -7/35 & -11/35 & 8/35 \end{bmatrix}$$

Finally, 2/3 times row 2 added to row 1 gives

$$\begin{bmatrix} 1 & 0 & 0 & 7/35 & 6/35 & 2/35 \\ 0 & 1 & 0 & -14/35 & -2/35 & 11/35 \\ 0 & 0 & 1 & -7/35 & -11/35 & 8/35 \end{bmatrix}$$

Therefore,

$$\underline{\underline{A}}^{-1} = \frac{1}{35}\begin{bmatrix} 7 & 6 & 2 \\ -14 & -2 & 11 \\ -7 & -11 & 8 \end{bmatrix}$$

which is the same result obtained from Method I (as expected).

A couple of convenient relationships that should also be noted are as follows:

1. If $\underline{\underline{A}}$ is a diagonal matrix (a square matrix with all zeros in the off-diagonal locations), then

$$\underline{\underline{A}}^{-1} = \left[a_{ii}^{-1} \right] \tag{3.22}$$

2. The inverse of a product of matrices is simply the product of the individual inverses in the opposite order, or

$$\left(\underline{\underline{A}}\,\underline{\underline{B}}\right)^{-1} = \underline{\underline{B}}^{-1}\underline{\underline{A}}^{-1} \tag{3.23}$$

To show this last relationship (as an example of manipulating matrix equations), we have

$$\left(\underline{\underline{A}}\,\underline{\underline{B}}\right)^{-1} = C$$

$$\left(\underline{\underline{A}}\,\underline{\underline{B}}\right)\left(\underline{\underline{A}}\,\underline{\underline{B}}\right)^{-1} = \underline{\underline{A}}\,\underline{\underline{B}}\,\underline{\underline{C}} = \underline{\underline{I}}$$

Thus $\underline{\underline{B}}\,\underline{\underline{C}} = \underline{\underline{A}}^{-1}$, or $\underline{\underline{C}} = \underline{\underline{B}}^{-1}\underline{\underline{A}}^{-1}$, which proves the above statement.

Rank of a Matrix

The ***rank*** of a matrix is defined as the maximum number of linearly independent rows or columns in the matrix. It is important to note that elementary row operations do not alter the rank of a matrix. Also it should be noted that $\underline{\underline{A}}$ and $\underline{\underline{A}}^T$ have the same rank.

Given m equations (i.e. m rows) with n unknowns (i.e. n columns),

$$\underline{\underline{A}}\,\underline{x} = \underline{b} \quad \text{or in augmented form} \quad \tilde{\underline{\underline{A}}} = \left[\underline{\underline{A}} \quad \underline{b} \right]$$

one can make the following statements concerning the existence and uniqueness of solutions for this matrix system:

1. This system has nontrivial solutions only if rank $\underline{\underline{A}}$ and rank $\tilde{\underline{\underline{A}}}$ are equal.

2. The system has precisely one solution if rank $\underline{\underline{A}}$ is n, where n is the number of unknowns in vector $\underline{x}$.

3. The system has infinitely many solutions if rank $\underline{\underline{A}}$ is less than n.

In general, if m > n, we have an over-determined system, and usually no solutions exist. If rank $\underline{\underline{A}}$ is m, then there are m linearly independent rows and only n unknowns; thus, there are no nontrivial solutions that can satisfy m independent equations with n unknowns (with n < m). In this case one usually uses the method of least squares to find the best solution based on some specific objective criterion (the reader is referred to the literature for further information on this subject).

If m < n, we have an under-determined system and usually an infinite number of solutions will satisfy these conditions.

If m = n (i.e. a square matrix) and rank $\underline{\underline{A}}$ = m = n, there is a single unique solution. This is normally the case of interest.

If $\underline{b} = 0$ (homogeneous system), then for a nontrivial solution, rank $\underline{\underline{A}}$ must be less than n. This says that the system matrix must have linearly dependent rows (which implies that the determinant is zero - see below).

The Case of n Equations and n Unknowns

For the usual case of n simultaneous equations with n unknowns, we have

$$\underline{\underline{A}}\,\underline{x} = \underline{b} \qquad \text{and} \qquad \underline{x} = \underline{\underline{A}}^{-1}\underline{b} \qquad \text{where} \qquad \underline{\underline{A}}^{-1} = \frac{\underline{\underline{C}}^{T}}{\det \underline{\underline{A}}}$$

Now two situations can occur:

I. Non-Homogeneous Problems:

In this case, $\underline{b} \neq 0$, and this system is said to be a non-homogeneous system. For this situation, there is only a single non-trivial solution if we have n linearly independent rows, which implies that rank $\underline{\underline{A}}$ = n, that the $\det \underline{\underline{A}}$ is nonzero, and that $\underline{\underline{A}}^{-1}$ exists. When $\underline{\underline{A}}^{-1}$ exists, $\underline{\underline{A}}$ is said to be non-singular.

II. Homogeneous Problems:

If $\underline{b} = 0$, then $\underline{x} = \underline{\underline{A}}^{-1}\underline{b}$ implies, at first glance, that $\underline{x}$ must be the null vector since we are multiplying the inverse matrix and the $\underline{b} = 0$ vector. However, if $\det \underline{\underline{A}} = 0$, then $\underline{\underline{A}}^{-1}$ does not exist, and the solution form, $\underline{x} = \underline{\underline{A}}^{-1}\underline{b}$, leads to an indeterminate form, which could lead to a nontrivial solution. In fact, this is indeed the situation, and we can argue that there are nontrivial solutions only if $\underline{\underline{A}}^{-1}$ does not exist. In this case we say that $\underline{\underline{A}}$ is a singular matrix. This happens only if $\det \underline{\underline{A}} = 0$ which implies that the rows of $\underline{\underline{A}}$ are linearly dependent and that rank $\underline{\underline{A}} < n$.

These conditions for homogeneous problems are very important in practice, and they lead naturally to the discussion of Eigenvalue-Eigenvector problems, which will be the next subject in this review unit on Linear Algebra.

Eigenvalue/Eigenvector Problems

Overview

From the discussion in the previous subsection on Systems of Linear Algebraic Equations, we saw that homogeneous equations with n equations and n unknowns require that the system matrix be singular for the existence of nontrivial solutions. The ***classical eigenvalue problem*** is a special case that falls into this class of problems and it arises from the general problem given by $\underline{\underline{A}}\underline{x} = \underline{b}$ when $\underline{b} = \lambda\underline{x}$ (that is, the right hand side vector is some constant times the solution vector $\underline{x}$). With this substitution, we have

$$\underline{\underline{A}}\underline{x} = \lambda\underline{x} \tag{3.24}$$

These systems occur frequently in applications and are usually written as

$$\left(\underline{\underline{A}} - \lambda\underline{\underline{I}}\right)\underline{x} = 0 \tag{3.25}$$

which is a homogeneous system of equations. Therefore, for non-trivial solutions, we require that

$$\det\left(\underline{\underline{A}} - \lambda\underline{\underline{I}}\right) = 0 \tag{3.26}$$

which is referred to as the characteristic equation. This gives rise to an n^{th} order polynomial in λ which has n roots -- the n eigenvalues of a square matrix of order n.

Note that the eigenvalues may be real and distinct, complex conjugates, repeated, or some combination of these. Note also that the sequence $\lambda_1, \lambda_2, \cdots \lambda_n$ is called the eigenvalue spectrum, with the magnitude of the largest eigenvalue denoted as the spectral radius, or

$$\left|\lambda_{max}\right| = \text{spectral radius}$$

The eigenvector $\underline{x}_i$ associated with the i^{th} eigenvalue, λ_i, is found by evaluating the homogeneous equation

$$\left(\underline{\underline{A}} - \lambda_i\underline{\underline{I}}\right)\underline{x}_i = 0 \tag{3.27}$$

An Example

As an example, let's find the eigenvalues and eigenvectors of the following matrix:

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

The characteristic equation is given by

$$\left|\underline{\underline{A}} - \lambda\underline{\underline{I}}\right| = \begin{vmatrix} 2-\lambda & -1 & 0 \\ -1 & 2-\lambda & -1 \\ 0 & -1 & 2-\lambda \end{vmatrix} = 0$$

Expanding the determinant along row 1 using Laplace's expansion gives

$$\left|\underline{\underline{A}} - \lambda \underline{\underline{I}}\right| = \lambda^3 - 6\lambda^2 + 10\lambda - 4 = 0$$

and the roots of this 3rd order polynomial are (obtained from Matlab)

$$\lambda_1 = 2 \quad \text{and} \quad \lambda_{2,3} = 2 \pm \sqrt{2}$$

Now, the eigenvector associated with the ith eigenvalue can be determined by solving the matrix equations with the specific eigenvalue inserted into the equation. For example, for $\lambda_1 = 2$, we have

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

which gives three equations

$$-x_2 = 0 \,, \quad -x_1 - x_3 = 0 \,, \quad \text{and} \quad -x_2 = 0$$

Clearly, the first and third equations are redundant, as expected. These equations, along with the second equation, which implies that $x_3 = -x_1$, identify the first eigenvector as

$$\underline{x}_1 = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

where we have chosen the first component to be unity. Clearly there is an arbitrary normalization associated with the eigenvector (because a homogeneous system of rank n-1 will always have an infinite number of solutions that simply differ by a single normalization parameter). One common practice is to normalize the magnitude of the vector to unity to force some specificity for the normalization constant. If this is done, the above eigenvector becomes

$$\hat{\underline{x}}_1 = \frac{\underline{x}_1}{\left|\underline{x}_1\right|} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

Both $\underline{x}_1$ and $\hat{\underline{x}}_1$ are valid eigenvectors for this eigenvalue (differ only by a normalization).

For $\lambda_2 = 2 + \sqrt{2}$, we can perform the same operations to get

$$\begin{bmatrix} -\sqrt{2} & -1 & 0 \\ -1 & -\sqrt{2} & -1 \\ 0 & -1 & -\sqrt{2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

or

$$-\sqrt{2}x_1 - x_2 = 0\ , \qquad -x_1 - \sqrt{2}x_2 - x_3 = 0\ , \qquad \text{and} \qquad -x_2 - \sqrt{2}x_3 = 0$$

and these give the relations

$$x_1 = \frac{-x_2}{\sqrt{2}} \qquad \text{and} \qquad x_3 = \frac{-x_2}{\sqrt{2}}$$

which says that $x_1 = x_3$ and x_2 is arbitrary (from the second equation in the original set). For convenience, we choose $x_2 = -\sqrt{2}$, which gives

$$\underline{x}_2 = \begin{bmatrix} 1 \\ -\sqrt{2} \\ 1 \end{bmatrix}$$

Similarly, the third eigenvector can be determined to be

$$\underline{x}_3 = \begin{bmatrix} 1 \\ \sqrt{2} \\ 1 \end{bmatrix}$$

using the same method as above.

Summary Note

The capability to do computations of this type is built directly into Matlab and other similar programs and, in practice, automated routines like those in Matlab (see the Matlab demo in a later subsection) are used in day-to-day engineering applications as needed. However, the student should definitely know the fundamentals of these numerical algorithms (although the details are not always necessary). By assuring that you can do the above manipulations by hand for low order systems, you will gain the confidence and experience necessary to intelligently and efficiently use the automated software. Thus, you should make sure you understand the above example, and be able to perform similar manipulations on small systems as verification of the computer tools that simply automate the procedures.

Some Special Matrices

Three Special Classes

There are a number of matrices that deserve some special mention. In particular, of interest here are three classes of matrices -- Hermitian, Skew-Hermitian, and Unitary Matrices.

Hermitian matrices satisfy the relationship

$$\tilde{\underline{\underline{A}}}^T = \underline{\underline{A}} \quad \text{or} \quad \tilde{a}_{ji} = a_{ij} \tag{3.28}$$

where the wavy line over an element implies that one should take the complex conjugate of that quantity. An example of a hermitian matrix is

$$\underline{\underline{A}} = \begin{bmatrix} 4 & 1-3i \\ 1+3i & 7 \end{bmatrix}$$

Note that the diagonal elements of hermitian matrices must be real, since $\tilde{a}_{ii} = a_{ii}$ for the diagonal elements.

Skew-hermitian matrices are similar to the above definition except for a negative relationship,

$$\tilde{\underline{\underline{A}}}^T = -\underline{\underline{A}} \quad \text{or} \quad \tilde{a}_{ji} = -a_{ij} \tag{3.29}$$

In this case, the diagonal elements must be pure imaginary, since $\tilde{a}_{ii} = -a_{ii}$ along the diagonal. This really says that $(\alpha - j\beta) = -(\alpha + j\beta)$, which clearly implies that $\alpha = 0$. As an example, consider the following matrix,

$$\underline{\underline{A}} = \begin{bmatrix} 3i & 2+i \\ -2+i & -i \end{bmatrix}$$

A ***unitary matrix*** is one that satisfies the expression

$$\tilde{\underline{\underline{A}}}^T = \underline{\underline{A}}^{-1} \tag{3.30}$$

As an example, if $\underline{\underline{A}}$ is given by $\underline{\underline{A}} = \begin{bmatrix} i/2 & \sqrt{3}/2 \\ \sqrt{3}/2 & i/2 \end{bmatrix}$ then $\tilde{\underline{\underline{A}}}^T = \begin{bmatrix} -i/2 & \sqrt{3}/2 \\ \sqrt{3}/2 & -i/2 \end{bmatrix}$ and $\underline{\underline{A}}^{-1}$ should be the same as $\tilde{\underline{\underline{A}}}^T$. As a check, let's compute the product $\tilde{\underline{\underline{A}}}^T \underline{\underline{A}}$ and see if it gives the identity matrix, or

$$\begin{bmatrix} -i/2 & \sqrt{3}/2 \\ \sqrt{3}/2 & -i/2 \end{bmatrix} \begin{bmatrix} i/2 & \sqrt{3}/2 \\ \sqrt{3}/2 & i/2 \end{bmatrix} = \begin{bmatrix} 1/4+3/4 & 0 \\ 0 & 3/4+1/4 \end{bmatrix} = \underline{\underline{I}}$$

Note that hermitian, skew-hermitian, and unitary matrices are, in general, complex matrices. However, a subset of each of these classes exists for the case of all real elements, and they go by the names ***symmetric***, ***skew-symmetric***, and ***orthogonal***, respectively.

The nice thing about matrices within these classes is that we can characterize their eigenvalue spectrum, as follows:

1. The eigenvalues of a hermitian (or real symmetric) matrix are real.
2. The eigenvalues of a skew-hermitian (or real skew-symmetric) matrix are imaginary or zero.
3. The eigenvalues of a unitary matrix (or real orthogonal) matrix have absolute value of unity.

For the examples given above we can compute the eigenvalues, giving

Hermitian matrix $\quad \lambda^2 - 11\lambda + 18 = 0 \quad$ and $\quad \lambda_{1,2} = 9,\ 2$

Skew-hermitian matrix $\quad \lambda^2 - 2i\lambda + 8 = 0 \quad$ and $\quad \lambda_{1,2} = 4i, -2i$

Unitary matrix $\quad \lambda^2 - i\lambda - 1 = 0 \quad$ and $\quad \lambda_1 = \frac{1}{2}\left(\sqrt{3} + i\right),\ \lambda_2 = \frac{1}{2}\left(-\sqrt{3} + i\right)$

where this last set of eigenvalues was obtained from Matlab. Also note that the magnitude of each eigenvalue for the unitary matrix is indeed unity. For example, the magnitude of λ_1 is given by

$$|\lambda_1| = \sqrt{\left(\frac{\sqrt{3}}{2}\right)^2 + \left(\frac{1}{2}\right)^2} = \sqrt{\left(\frac{3}{4} + \frac{1}{4}\right)} = 1$$

Quadratic Forms

The form $\underline{x}^T \underline{\underline{A}}\,\underline{x}$ is a common combination of terms that occurs frequently in applications. In particular, if $\underline{\underline{A}}$ and $\underline{x}$ are both real, then the combination $\underline{x}^T \underline{\underline{A}}\,\underline{x}$ is referred to as a ***quadratic form***. Also if $\underline{\underline{A}}$ is hermitian, then $\underline{x}^T \underline{\underline{A}}\,\underline{x}$ is real for any $\underline{x}$, and if $\underline{\underline{A}}$ is skew-hermitian, then $\underline{x}^T \underline{\underline{A}}\,\underline{x}$ is pure imaginary for any $\underline{x}$. These summary properties can be useful in some cases.

Similar Matrices

One final set of terminology related to the eigenvalues of a matrix still needs to be discussed – that is the concept of ***similar matrices*** and related subjects. In particular, two matrices, $\underline{\underline{A}}$ and $\underline{\underline{B}}$, are said to be ***similar*** if they satisfy the relation

$$\underline{\underline{B}} = \underline{\underline{T}}^{-1} \underline{\underline{A}}\,\underline{\underline{T}} \tag{3.31}$$

where $\underline{\underline{T}}$ is a transformation matrix. This transformation is said to be a ***Similarity Transformation***. The important point here is that ***similar matrices have the same eigenvalues***. In addition, if $\underline{x}$ is an eigenvector of $\underline{\underline{A}}$, then $\underline{y} = \underline{\underline{T}}^{-1}\underline{x}$ is the eigenvector of $\underline{\underline{B}}$ corresponding to that same eigenvalue. We can show this by the following manipulations:

$$\underline{\underline{A}}\,\underline{x} = \lambda \underline{x}$$

$$\underline{\underline{T}}^{-1} \underline{\underline{A}}\,\underline{x} = \lambda \underline{\underline{T}}^{-1} \underline{x}$$

$$\underline{\underline{T}}^{-1}\underline{\underline{A}}\underline{\underline{T}}\underline{\underline{T}}^{-1}\underline{x} = \lambda\underline{\underline{T}}^{-1}\underline{x}$$

$$\underline{\underline{B}}\underline{\underline{T}}^{-1}\underline{x} = \lambda\underline{\underline{T}}^{-1}\underline{x}$$

$$\underline{\underline{B}}\underline{y} = \lambda\underline{y}$$

This set of expressions shows that λ is indeed an eigenvalue of $\underline{\underline{B}}$ and that $\underline{y} = \underline{\underline{T}}^{-1}\underline{x}$ is the eigenvector of $\underline{\underline{B}}$ that corresponds to eigenvalue λ.

If $\lambda_1, \lambda_2, \cdots \lambda_n$ are distinct eigenvalues of an n×n matrix, then the corresponding eigenvectors $\underline{x}_1, \underline{x}_2, \underline{x}_3, \cdots \underline{x}_n$ form a linearly independent set and they represent a basis of eigenvectors in n dimensional space.

The ***modal matrix*** is a special matrix whose columns contain the linearly independent eigenvectors/basis vectors, or

$$\underline{\underline{M}} = \begin{bmatrix} \underline{x}_1 & \underline{x}_2 & \cdots & \underline{x}_n \end{bmatrix} \tag{3.32}$$

Also we should note that any vector, $\underline{y}$, has a unique representation in n dimensional space simply as a linear combination of the basis vectors, or

$$\underline{y} = c_1\underline{x}_1 + c_2\underline{x}_2 + c_3\underline{x}_3 + \cdots c_n\underline{x}_n \tag{3.33}$$

Also note that a ***linear transformation***, $\underline{z} = \underline{\underline{A}}\underline{y}$, in terms of the basis vectors, becomes

$$\underline{z} = \underline{\underline{A}}\underline{y} = \underline{\underline{A}}\left[c_1\underline{x}_1 + c_2\underline{x}_2 + c_3\underline{x}_3 + \cdots c_n\underline{x}_n\right]$$

or

$$\underline{z} = c_1\lambda_1\underline{x}_1 + c_2\lambda_2\underline{x}_2 + \cdots c_n\lambda_n\underline{x}_n \tag{3.34}$$

Now if we let the transformation matrix, $\underline{\underline{T}}$, in the above similarity transformation expression [see eqn. (3.31)] be composed of the basis vectors for the n-dimensional problem, or

$$\underline{\underline{T}} = \underline{\underline{M}} = \begin{bmatrix} \underline{x}_1 & \underline{x}_2 & \cdots & \underline{x}_n \end{bmatrix}$$

then,

$$\underline{\underline{M}}^{-1}\underline{\underline{A}}\underline{\underline{M}} = \underline{\underline{D}} \tag{3.35}$$

which is a diagonal matrix with the eigenvalues of $\underline{\underline{A}}$ along the diagonal of $\underline{\underline{D}}$. Also, note that a similar relationship that is often used is

$$\underline{\underline{A}} = \underline{\underline{M}}\underline{\underline{D}}\underline{\underline{M}}^{-1} \tag{3.36}$$

A proof of the first relationship can be demonstrated as follows:

$$\underline{\underline{A}}\underline{\underline{M}} = \underline{\underline{A}}\begin{bmatrix} \underline{x}_1 & \underline{x}_2 & \cdots & \underline{x}_n \end{bmatrix} = \begin{bmatrix} \lambda_1\underline{x}_1 & \lambda_2\underline{x}_2 & \cdots & \lambda_n\underline{x}_n \end{bmatrix}$$

$$\underline{\underline{A}}\,\underline{\underline{M}} = \begin{bmatrix} \underline{x}_1 & \underline{x}_2 & \cdots & \underline{x}_n \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \lambda_n \end{bmatrix}$$

or $\underline{\underline{A}}\,\underline{\underline{M}} = \underline{\underline{M}}\,\underline{\underline{D}}$ and $\underline{\underline{M}}^{-1}\underline{\underline{A}}\,\underline{\underline{M}} = \underline{\underline{D}}$ as given above.

A Matlab Demo

A short Matlab demo has been prepared to illustrate some of the matrix/vector operations that can be performed within this programming environment. A few quantities are defined

$$\underline{x} = \begin{bmatrix} 2 \\ 1 \\ -1 \end{bmatrix} \qquad \underline{y} = \begin{bmatrix} 0 \\ -4 \\ 3 \end{bmatrix} \qquad \underline{\underline{A}} = \begin{bmatrix} 1 & 0 & 5 \\ 0 & 4 & -2 \\ 1 & 3 & 2 \end{bmatrix} \qquad \underline{\underline{B}} = \begin{bmatrix} 6 & -2 & 2 \\ 8 & 3 & 2 \end{bmatrix}$$

and then several manipulations are performed with these variables and related quantities. The Matlab commands are stored in the file **linademo.m** and a diary file is created during the run and saved as **linademo.out**. The actual m-file has several comments and it also displays (via the ***disp*** command) various comments to the output diary file. With this information, the Matlab file is very easy to follow and is quite self-explanatory. These files, the main program **linademo.m** and the diary file **linademo.out** are listed in Table 3.1 and Table 3.2, respectively.

This demo only touches on a few of the matrix operations that can be accomplished with the Matlab package. If you are a new Matlab user, you should use this as a start, and then build upon this capability as needed for specific applications. The Matlab documentation is a good source of information as you become more experienced and your needs grow. Also, as mentioned previously, I have prepared a variety of Matlab demos for my undergraduate courses -- especially Differential Equations and Applied Problem Solving with Matlab -- and these can be accessed over the web at www.profjrwhite.com/courses.htm. I think you will find that Matlab is extremely powerful and quite easy to use for typical applications.

Table 3.1 Listing of Matlab file linademo.m.

```
%
%   LINADEMO.M    Linear Algebra Applications in MATLAB
%
%   This file simply demonstrates several linear algebra operations that
%   can be performed in Matlab.  This area is, in fact, one of Matlab's
%   greatest strengths, since the base element is a 2-d array and all
%   operations are, by default, matrix manipulations. In addition to the
%   basic arithmetic operations, Matlab also has m-files for just about
%   any matrix application or operation you can imagine.  This demo just
%   illustrates a few of Matlab's capabilities.
%
%   File prepared by J. R. White, UMass-Lowell  (July 2003).
%

%
%   Getting started
      clear all, close all
%
%   Open diary file for saving solutions
      delete linademo.out
      diary linademo.out
      format compact
      disp(' *** LINADEMO.OUT ***   Diary File for LINADEMO.M ')
      disp(' ')
%
%   Define some matrices for the sample problems which follow
      disp('Matrices for sample manipulations')
      x = [2 1 -1]',       y = [0 -4 3]'
      A = [1 0 5; 0 4 -2; 1 3 2],      B = [6 -2 2; 8 3 2]
```

```
%
%   Now let's perform several arithmetic operations
      disp('Find inner product of x and y');                    xx = x'*y
      disp('Find outer product of x and y');                    XX = x*y'
      disp('Try getting the row and column dimensions of XX');  [nr,nc] = size(XX)
      disp('Note that A*B is undefined, but we can do B*A');    C = B*A
      disp('What is the size of B transposed * B?');            size(B'*B)
      disp('How about the size of B * B transposed?');          size(B*B')
%
%   We can also extract portions of a matrix (using the repeat operator (the :) )
      disp('The first column of A is');                           a1 = A(:,1)
      disp('Or the first and third rows of A can be extracted');  a13 = A(1:2:3,:)
%
%   Working with systems of equations is also easy
      disp('The rank of A is');            rankA = rank(A)
      disp('The determinant of A is');     detA = det(A)
      disp('The inverse of A is');         invA = inv(A)
      disp('The solution to Az = y can be found as z = invA*y');      z1 = invA*y
      disp('Or, more efficiently, with an LU decomposition scheme');  z2 = A\y
      disp('We can see the components of the LU decomp scheme');      [L,U,P] = lu(A)
      disp('With this form, we should have L*U = P*A, or L*U - P*A = 0'); ZZ = L*U-P*A
%
      disp(['Also if you do not know how to use a command, just type ' ...
           '"help command name".']);
      disp('For example, help lu gives');  help lu
%
%   Finding eigenvalues and eigenvectors is also straightforward
      disp('The eigenvalues & eigenvectors of A are');      [M,D] = eig(A)
      disp(['For distinct eigenvalues, M should satisfy the similarity ' ...
            'transformation, D = invM*A*M.  Let us see!']);  DD = inv(M)*A*M
%
%   Well this demo could go on and on, so let's finish by simply closing the diary file
      disp('And so on, and so on, and so on, ...');
      diary off
%
%   end of demo
```

Table 3.2 Listing of diary file linademo.out from linademo.m.

```
*** LINADEMO.OUT ***   Diary File for LINADEMO.M

Matrices for sample manipulations
x =
     2
     1
    -1
y =
     0
    -4
     3
A =
     1     0     5
     0     4    -2
     1     3     2
B =
     6    -2     2
     8     3     2
Find inner product of x and y
xx =
    -7
Find outer product of x and y
XX =
     0    -8     6
     0    -4     3
     0     4    -3
Try getting the row and column dimensions of XX
nr =
     3
nc =
     3
```

```
Note that A*B is undefined, but we can do B*A
C =
     8    -2    38
    10    18    38
What is the size of B transposed * B?
ans =
     3     3
How about the size of B * B transposed?
ans =
     2     2
The first column of A is
a1 =
     1
     0
     1
Or the first and third rows of A can be extracted
a13 =
     1     0     5
     1     3     2
The rank of A is
rankA =
     3
The determinant of A is
detA =
    -6
The inverse of A is
invA =
   -2.3333   -2.5000    3.3333
    0.3333    0.5000   -0.3333
    0.6667    0.5000   -0.6667
The solution to Az = y can be found as z = invA*y
z1 =
    20
    -3
    -4
Or, more efficiently, with an LU decomposition scheme
z2 =
    20
    -3
    -4
We can see the components of the LU decomp scheme
L =
    1.0000         0         0
         0    1.0000         0
    1.0000    0.7500    1.0000
U =
    1.0000         0    5.0000
         0    4.0000   -2.0000
         0         0   -1.5000
P =
     1     0     0
     0     1     0
     0     0     1
With this form, we should have L*U = P*A, or L*U - P*A = 0
ZZ =
     0     0     0
     0     0     0
     0     0     0
Also if you do not know how to use a command, just type "help command name".
For example, help lu gives

 LU     LU factorization.
    [L,U] = LU(X) stores an upper triangular matrix in U and a
    "psychologically lower triangular matrix" (i.e. a product
    of lower triangular and permutation matrices) in L, so
    that X = L*U. X can be rectangular.

    [L,U,P] = LU(X) returns unit lower triangular matrix L, upper
    triangular matrix U, and permutation matrix P so that
    P*X = L*U.

    Y = LU(X) returns the output from LAPACK'S DGETRF or ZGETRF
    routine if X is full. If X is sparse, Y contains the strict
    lower triangle of L embedded in the same matrix as the upper
    triangle of U. In both full and sparse cases, the permutation
```

```
    information is lost.

    [L,U,P,Q] = LU(X) returns unit lower triangular matrix L,
    upper triangular matrix U, a permutation matrix P and a column
    reordering matrix Q so that P*X*Q = L*U for sparse non-empty X.
    This uses UMFPACK and is significantly more time and memory
    efficient than the other syntaxes, even when used with COLAMD.

    [L,U,P] = LU(X,THRESH) controls pivoting in sparse matrices,
    where THRESH is a pivot threshold in [0,1].  Pivoting occurs
    when the diagonal entry in a column has magnitude less than
    THRESH times the magnitude of any sub-diagonal entry in that
    column.  THRESH = 0 forces diagonal pivoting.  THRESH = 1 is
    the default.

    [L,U,P,Q] = LU(X,THRESH) controls pivoting in UMFPACK, where
    THRESH is a pivot threshold in [0,1].  Given a pivot column j,
    UMFPACK selects the sparsest candidate pivot row i such that
    the absolute value of the pivot entry is greater than or equal
    to THRESH times the largest entry in the column j.  The magnitude
    of entries in L is limited to 1/THRESH.  A value of 1.0 results
    in conventional partial pivoting.  The default value is 0.1.
    Smaller values tend to lead to sparser LU factors, but the
    solution can become inaccurate.  Larger values can lead
    to a more accurate solution (but not always), and usually an
    increase in the total work.

    See also COLAMD, LUINC, QR, RREF, UMFPACK.

 Overloaded methods
    help gf/lu.m

The eigenvalues & eigenvectors of A are
M =
  -0.9590            -0.7297            -0.7297
   0.1186             0.2224 - 0.4244i   0.2224 + 0.4244i
   0.2575            -0.3898 - 0.2932i  -0.3898 + 0.2932i
D =
  -0.3426                  0                  0
        0             3.6713 + 2.0092i        0
        0                  0             3.6713 - 2.0092i
For distinct eigenvalues, M should satisfy the similarity transformation, D = invM*A*M.  Let us
see!
DD =
  -0.3426 - 0.0000i   0.0000 + 0.0000i   0.0000 - 0.0000i
  -0.0000 + 0.0000i   3.6713 + 2.0092i  -0.0000 - 0.0000i
  -0.0000 - 0.0000i  -0.0000 + 0.0000i   3.6713 - 2.0092i
And so on, and so on, and so on, ...
```