Linear algebraic equations and matrices

Maths 235 G. Keady, University of Western Australia

LINEAR ALGEBRA/ MATRIX ALGEBRA IS IMPORTANT

The engineering examples in the table below are those where the linear algebra application is immediately obvious. They mostly come from easier application areas, including easy ones from Mech Eng, etc.

<table>
<thead>
<tr>
<th>HDR</th>
<th>ON</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ch3</td>
<td>Ch7</td>
<td>vectors in geometry and mechanics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>matrices for rotations, etc. in geometry (incl. computer graphics)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>moment of inertia matrix in mechanics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>least squares approximation of data, regression</td>
</tr>
</tbody>
</table>

Mathematical uses
- treating ...
- so that it looks like ...
- multivariable calculus
- 1-variable calculus
- \( Df = \left( \frac{\partial f_i}{\partial x_j} \right) \)
- jacobian
- classification of extrema
- e.g. minima
- \( \left( \frac{\partial^2 f}{\partial x_i \partial x_j} \right) \geq 0 \)
- in a sense we will define later
- (See eigenvalues: symmetric matrices)

<table>
<thead>
<tr>
<th>HDR</th>
<th>ON</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>§5.6-5.9</td>
<td>§10.5</td>
<td>linear systems of d.e.s.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>single linear d.e.</td>
</tr>
<tr>
<td>§5.6-5.9</td>
<td>§10.5</td>
<td>( \frac{dy}{dt} = Ay )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \frac{dy}{dt} = A_y )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( y(t) = \text{exponential}(A,t)y(0) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( y(t) = \text{exp}(At)y(0) )</td>
</tr>
</tbody>
</table>

Here are some engineering applications involving matrices.
- Statics, pin-jointed frameworks (first year)
- Elasticity (finite elements in later mech. and civil eng. courses)
- Vibration problems: calculation of resonant frequencies, occurring in mech. and electrical applications.
- Electrical networks - linear circuits
  - steady
  - transient - linear d.e. (See section on eigenvalues.)
- Hydraulic networks - water distribution pipe networks; or nonlinear electrical networks
  - steady, this time a nonlinear problem but solved by iteration involving local linearisation.
  - transient - nonlinear d.e. (Do it numerically!)
In the last, ‘network’ cases the ‘steady’ (or ‘equilibrium’) case is the most closely related to easy
uses of matrices.

There are two aspects to the subject of linear algebra: theory and practical calculations. Both
are important, though as engineers I expect that you will take to the practical calculations
side of the subject first, and will only learn the theory when you feel the need. This is actually
inefficient (though understandable): if I refer to theory items, please trust me that they will be
useful.

LINEAR SYSTEMS $Ax = b$ (ON §8.1)

There is no mathematics in this section which you haven’t already met in your first year linear
algebra. The summary of the section is that, in engineering applications, you can find many
examples involving solving linear systems.

Introduction to practical calculations in $\mathbb{R}^n$

Consider solving linear systems of equations

$$
\sum_j a_{ij} x_j = b_i.
$$

In matrix notation, this is written $Ax = b$: there will be much more on how matrices help later.

In first year you learnt about solving systems of linear equations by Gauss elimination, or
equivalently, row equivalent systems (ON §8.3, 8.7). Later in this course we will mention $LU$
-factorisation, which is the name given to the same method when it is implemented for hardware-float
computations.

The problem of solving linear systems $Ax = b$ is an easy, standard problem. There are lots of
calculator programs to solve it, e.g.
Maple $\text{llinsolve}(A, b)$, Matlab $A \backslash b$

Important task for you: stay awake in your engineering lectures. Look for the places where
linear equations arise.

Next we give some details of a couple of typical easy engineering examples which lead to solving
linear systems of equations.

Example from DC-circuitry: Wheatstone bridge circuit

(See [UWA1] §1.1 Example 1.2. Some related circuit topics are treated in HDR Chpt 3 p104. For
a more general account, see [Str] in the early parts of Chpt 8.)

In Figure 1 we have drawn the Wheatstone bridge circuit.

Given: resistances, $R_j$, $j = 1..5$, and voltage $V$ of battery.
Find: distribution of currents $I_j$, current $I_t$ flowing through resistor $R_t$.

The equations can be set up as follows. Apply Kirchhoff’s rule for conservation of charge at the
nodes. This gives

$$
I_1 + I_2 - I_t = 0,
I_1 - I_3 + I_5 = 0,
I_3 + I_4 - I_t = 0.
$$
Apply Kirchhoff’s rule for voltage drops around closed loops. This gives

\[ R_1 I_1 + R_3 I_3 = V, \]
\[ R_1 I_1 - R_2 I_2 - R_5 I_5 = 0, \]
\[ R_3 I_3 - R_4 I_4 + R_5 I_5 = 0. \]

The preceding give a system of 6 linear equations in the 6 unknown currents. Your mathematical software will have no difficulty solving them. (Exercise for you. Get your CAS to solve them and then to find the relation between \( R_j \), \( j = 1..4 \), so that \( I_5 \) is zero.)

One of the practical uses of the circuit is to determine, using three known resistors \( R_1, R_2, R_3 \), including at least one adjustable calibrated resistor, the value of an unknown resistor \( R_4 \). A way to do this is to adjust the adjustable resistor(s) so that current \( I_5 \) showing on an ammeter is zero. The relation which effects this is

\[ R_4 = \frac{R_2 R_3}{R_1} \]

for \( I_5 = 0 \),

which you will discover if you do the exercise in the preceding paragraph.

**Example from statics: just-rigid frames**

There are many similar examples in textbooks like Meriam and Kraige *Statics* (Wiley, and in this book the examples are in the Chapter on Structures).

Given: applied loads. See Figure 2.

Find: forces acting along bars and external reactions at supports.
In just-rigid frames the bars are rigid and inextensible, and the joints are smoothly hinged. A frame is just-rigid when the removal of any one of its bars destroys its rigidity.

In an engineering statics course, you might have seen that: a just-rigid frame with \( j \) joints has \((2j - 3)\) bars. (The proof is essentially by setting up equations and counting equations and unknowns.)

In the diagram above there are 7 joints and \((14 - 3) = 11\) bars.

In the diagram below (Figure 3) there are \(2k + 1 = j\) vertices or joints, and the edges count as follows:

\[
\begin{align*}
&k \text{ bottom edges} \\
&k - 1 \text{ top edges} \\
&2k \text{ edges} \\
\end{align*}
\]

\[
4k - 1 \quad (2j - 3).
\]

The method of joints for solving for the stress in the \((2j - 3)\) bars and the external reactions at the 2 supports (3 unknown components) involves writing down the equations for the equilibrium of forces at each of the \( j \) joints: \(2j\) linear equations for the \((2j - 3) + (3)\) unknowns.

How to write down the equations? You might revise your statics. The bar is in equilibrium under 2 forces, the reactions at its ends. These 2 forces must be equal in magnitude, act in opposite senses along bar.
The arrows above indicate forces exerted on the joints by the bar.
For a bar in tension forces act so that bar tends to be torn in two.
For a bar in thrust (or compression) forces act so that bar tends to buckle.

EXAMPLE. A very simple truss. Symmetrically loaded equilateral triangle: Figure 4.

Figure 4: $\theta_1 = \theta_3 = \pi/3$ in this example.

At 1,
\[
T_{13} + \frac{1}{2}T_{12} - X = 0, \\
\frac{\sqrt{3}}{2}T_{12} - Y_1 = 0,
\]

At 3,
\[
-T_{13} - \frac{1}{2}T_{23} = 0, \\
\frac{\sqrt{3}}{2}T_{23} - Y_3 = 0,
\]

At 2,
\[
\frac{1}{2}T_{12} - \frac{1}{2}T_{23} = 0, \\
\frac{\sqrt{3}}{2}T_{12} + \frac{\sqrt{3}}{2}T_{23} = F.
\]

Solution. Adding $x$-component’s equations at 1 and 3 gives $X = 0$.

From equations 2, $T_{12} = T_{23} = \frac{F}{\sqrt{3}}$.

From $y$-component’s equations at 1 or 3 $Y_1 = Y_3 = \frac{F}{2}$.

Finally, $T_{13} = -\frac{1}{2}T_{12} = -\frac{F}{2\sqrt{3}}$.

Bars (1, 2) and (2, 3) are in thrust, (1, 3) in tension.

The system can be written $Ax = f$ where
\[
A = \begin{bmatrix}
1 & 0 & -\frac{1}{2} & -1 & 0 & 0 \\
0 & 1 & -\frac{\sqrt{3}}{2} & 0 & 0 & 0 \\
0 & 0 & \frac{1}{2} & 1 & -\frac{1}{2} & 0 \\
0 & 0 & \frac{\sqrt{3}}{2} & 0 & \frac{\sqrt{3}}{2} & 0 \\
0 & 0 & 0 & 1 & \frac{1}{2} & 0 \\
0 & 0 & 0 & 0 & -\frac{\sqrt{3}}{2} & 1
\end{bmatrix}, \quad x = \begin{bmatrix} X_1 \\ Y_1 \\ T_{12} \\ T_{13} \\ T_{23} \\ Y_3 \end{bmatrix}, \quad f = \begin{bmatrix} 0 \\ 0 \\ T_{12} \\ T_{13} \\ T_{23} \\ F \end{bmatrix}.
\]

In this the equations are in order $x-$, $y-$ components resp. of vertices 1, 2, 3, resp.
We will see later that \( A \) is a banded matrix, and that there are efficient numerical methods for banded matrices.

There are, no doubt, computer packages which would solve all truss/statics problems like this. Harder EXAMPLE. In 2nd year eng. maths assignments in some years, Mech./Civil/etc. students consider the truss in Figure 5.

![Figure 5: Single load example.](image)

Given: load (e.g. single load \( F_3 \) acting vertically down at joint 3),
Find: external reactions at supports \( X_1, Y_1, Y_7 \) and stresses \( T_{ij} \) in the 11 bars.

There are 14 linear equations in 14 unknowns. Let

\[
x = [X_1, Y_1, T_{12}, T_{13}, T_{23}, T_{24}, T_{34}, T_{35}, T_{45}, T_{56}, T_{57}, T_{67}, Y_7]^T,
\]

where the notation, superscript \( T \), is said transpose, and means that the \( x \) is a column vector, or an \( n \) by 1 matrix. In matrix form the system is

\[
Ax = f.
\]

(In the example, e.g. load, above \( f \) is the vector with 0 in every entry except the 6th.)

Consider the equations in order \( x-\), \( y-\) components for each of the joints 1 . . . 7 in that order.

The matrix \( A \) is ‘sparse’, meaning it ‘has lots of 0s’. Even better, with the ordering of the equations as suggested, \( A \) is ‘banded’,

\[
\begin{bmatrix}
** & 0 & & & & & & \\
* & ** & & & & & & \\
0 & * & ** & & & & & \\
& & & *** & & & & \\
& & & & *** & & & \\
& & & & & *** & & \\
& & & & & & & **
\end{bmatrix},
\]

meaning it ‘has its nonzero elements banded around the diagonal’.

Numerical solution is easier for banded systems.

**Hooke’s law and elastic pin-jointed frameworks**

This example will not be developed in the 2nd year Engineering Maths course. The pictures one draws in this application area look like those in the preceding subsection. Again it is statics. The difference how is that the bars are elastic, and stretch or compress according to the forces at the joints. Hooke’s law ensures that we deal with linear problems. There are treatments of the problem area, a little more advanced than this unit, in Strang’s *Introduction to Applied Maths* Chpt 2, and in Noble and Daniels *Applied Linear Algebra* Chpt 2.
The 2nd year Engineering Maths courses extend and build on your first year maths courses, here your first year linear algebra course. You should now revise ON §8.1, §8.3, [Str] Chpt 2-3 or HDR Chpt 3.

\( K \) will denote any field, but for our purposes it suffices to consider
\( K = \mathbb{Q} \) the rational numbers, or
\( K = \mathbb{R} \) the real numbers, or
\( K = \mathbb{C} \) the complex numbers.

**Definition.** (ON Defn 8.1, [Str] 2nd and 3rd ed. p27.) An \( m \times n \) matrix \( A = (a_{ij})_{m \times n} \) is a rectangular array of elements of \( K \) written
\[
A = \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
 a_{m1} & a_{m2} & \cdots & a_{mn}
\end{bmatrix}.
\]

**CAUTION.** [ON] uses \( n \) by \( m \) where I write \( m \) by \( n \). We both mean the first number to be the number of rows, and the second to be the number of columns. For both of us:

\[
\text{matrix size} = (\text{number of rows}) \times (\text{number of columns}) .
\]

Fortunately, we mostly deal with square matrices, and we both call them \( n \) by \( n \).

**Remarks.** (i) A notation which is becoming popularized by the spread of Object-Oriented Programming - OOP - (including OOP-style CAS like MuPAD, AXIOM, ...) is to denote the set of all \( m \) by \( n \) matrices with elements in \( K \) by \( \text{Matrix}(m, n, K) \) and \( \text{SquareMatrix}(n, K) = \text{Matrix}(n, n, K) \).

(ii) It makes sense – and has uses – to define matrices more generally than we have above, with elements not necessarily restricted to having to come from fields \( K \). A more general algebraic structure than field which is appropriate in this context is that of a ring. Thus people define \( \text{Matrix}(m, n, R) \) for a ring \( R \). (We won’t give the definition of ‘ring’ here, but give a couple of examples. The set \( UP(K) \) of all univariate polynomials with coefficients from field \( K \) forms a ring. The set \( \text{SquareMatrix}(n, K) \) with entries from field \( K \) forms a ring.)

(iii) Neither your text books, nor the older-style StudentMatlab/Maple/Mathematica CAS you use, use this ‘Domain Constructor’ notation of (i), so I have refrained from using it much in these notes.

**THEOREM.** (ON Thm 8.1.) The set of \( m \times n \) matrices over \( K \) forms a vector space (of dimension \( mn \)) over \( K \).

Both for setting up linear systems \( Ax = b \) and for other uses we now define matrix multiplication. The set-up is
\[
A \in \ell \times m, \quad B \in m \times n \quad \text{and we will define} \quad C = AB \in \ell \times n.
\]

**Definition.** (ON Defn 8.5.) Let \( A = (a_{ij})_{\ell \times m}, \quad B = (b_{ij})_{m \times n} \). Define the \((i, k)\)–th element of \( AB \) to be \( \sum_{j=1}^{m} a_{ij}b_{jk} \). The \( a_{ij} \) are from the \( i \)-th row of \( A \): the \( b_{jk} \) are from the \( k \)-th column of \( B \).

**THEOREM.** (ON Thm 8.1.) When the matrices have sizes so that the operations are defined
\[
A(B + C) = AB + AC, \quad (A + B)C = AC + BC, \quad A(BC) = (AB)C.
\]
Here are some more properties of matrix multiplication.
For a square matrix $A$ and positive integers $r$, $s$, (i) $A^r A^s = A^{r+s}$, (ii) $(A^r)^s = A^{rs}$.
For scalar $\lambda$ and matrices $A$, $B$, $A(\lambda B) = (\lambda A)B = \lambda(AB)$.
The matrix product $AB$ may be zero when neither $A$ nor $B$ is a zero matrix.

**Uses and examples of matrix multiplication**

A use we have already seen is the compact notation for writing a system of linear equations $Ax = b$, $m$ equations in $n$ unknowns:

$$
A = (a_{ij})_{m \times n},
\quad x = \text{column}_\text{vector}(x_1, \ldots, x_n),
\quad b = \text{column}_\text{vector}(b_1, \ldots, b_m).
$$

An EXAMPLE of matrix multiplication. Rotation through angle $\theta$. See HDR p134. (You have seen related geometric applications at the end of your first year linear algebra course ([UWA1] §4.4) in the section on applications to computer graphics. Material related to your first year computer graphics application is in [HDR] Chpt 3 p137, [Str] 2nd ed. §8.5, 3rd ed. §8.6.)

Consider matrix $M(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$. Now the example.

$$
M(\theta)M(\alpha) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} = \begin{bmatrix} \cos(\theta + \alpha) & -\sin(\theta + \alpha) \\ \sin(\theta + \alpha) & \cos(\theta + \alpha) \end{bmatrix}
$$

using school trig.,

$$
= M(\theta + \alpha) \text{ agreeing with what we expect geometrically.}
$$

Also $M(0) = I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and we will soon see definitions of inverse and that this says $M(-\theta)$ is the inverse of $M(\theta)$.

Matrix multiplication can be non-commutative.

**EXAMPLE.**

$$
R = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad M = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}
$$

reflection \hspace{1cm} \text{rotation through $\frac{\pi}{2}$}.

We have

$$
RM = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \text{ while } MR = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.
$$
EXAMPLE: 3-dimensional rotations through 90°.
(See also [HDR] Chpt 3 p135-.)

\[ M_z = \begin{bmatrix}
0 & -1 & 0 \\
1 & 0 & 0 \\
0 & 0 & 1
\end{bmatrix} \] rotation about \( z \)-axis,
\[ M_x = \begin{bmatrix}
1 & 0 & 0 \\
0 & 0 & -1 \\
0 & 1 & 0
\end{bmatrix} \] rotation about \( x \)-axis.

We have
\[ M_z M_x = \begin{bmatrix}
0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0
\end{bmatrix} \] while
\[ M_x M_z = \begin{bmatrix}
0 & -1 & 0 \\
0 & 0 & -1 \\
1 & 0 & 0
\end{bmatrix}. \]

There are many other simple examples and applications involving matrices and matrix multiplication. (For combinatorial and graph theoretic applications, see [ON] §8.2, [Str] in the early parts of Chpt 8. These, however, are not in the 2nd year Engineering Maths syllabus.)

Matrix inverses, properties. (ON §8.8)

(Alternative references include [Str] §2.4 or [HDR] Chpt 3.)

Definition. The \( n \) by \( n \) matrix \( I_n = [\delta_{ij}] \) with \( \delta_{ij} = 0 \) for \( i \neq j \) and \( \delta_{ii} = 1 \), i.e. zeros everywhere except on the diagonal where entries are 1, is called the \( n \) by \( n \) identity matrix.

(ON Defn 8.13 [Str] §2.5 2nd ed. p66, 3rd ed. p71.) A square matrix \( A \), say \( n \) by \( n \), is said to be invertible (or, in the alternative word of [ON] p373, nonsingular) if and only if there exists a matrix \( B \) such that
\[ AB = BA = I_n, \]
where \( I_n \) denotes the \( n \) by \( n \) identity matrix.

When the equations above are satisfied, \( B \) is called an inverse of \( A \).

Definition. A square matrix \( A \) is called singular if \( Ax = 0 \) has a nonzero solution for \( x \).

(We will soon see that nonsingular means not singular.)

THEOREM. (ON Thm 8.20.) An invertible matrix has exactly one inverse.

Proof. Suppose \( B \) and \( C \) are inverses of \( A \). Then
\[ B = BI = B(AC) = (BA)C = IC = C. \]
If \( B \) is an inverse of \( A \), we write \( B = A^{-1} \).

**THEOREM.** If \( A \) is invertible, then \( A \) is not singular.

Proof. Multiply, on the left, \( Ax = 0 \) by \( A^{-1} \).

\[
Ax = 0 \Rightarrow A^{-1}(Ax) = A^{-1}0 \Rightarrow x = 0.
\]

We remark that the converse is proved in ON Thm 8.23(2).

Finding a matrix inverse can be treated like solving several linear systems. (Later we will consider better ways of finding inverses.) Denote the \( i \)-th column of \( I_n \) by \( e_i \) and the \( i \)-th column of \( B \) by \( \text{col}(B, i) \). Then one can determine \( \text{col}(B, i) \) by solving

\[
A\text{col}(B, i) = e_i.
\]

Consider the linear system of equations \( Ax = b \). If we know \( A^{-1} \) the solution is \( x = A^{-1}b \).

**Important.** This is NOT used computationally with large systems as it is inefficient.

**EXAMPLES**

\[
M(\theta) = \begin{bmatrix}
\cos \theta & -\sin \theta \\
\sin \theta & \cos \theta
\end{bmatrix},
M(\theta)^{-1} = M(-\theta).
\]

\[
A = \begin{bmatrix}
a & b \\
c & d
\end{bmatrix}.
\]

If \( \delta = ad - bc \neq 0 \) then \( A^{-1} = \frac{1}{\delta} \begin{bmatrix}
d & -b \\
-c & a
\end{bmatrix} \).

**THEOREM.** (ON Thm 8.21.) If \( A \) and \( B \) are invertible matrices of the same size, then \( AB \) is invertible, and

\[
(AB)^{-1} = B^{-1}A^{-1}.
\]

Proof. \((AB)(B^{-1}A^{-1}) = A(BB^{-1})A^{-1} = AA^{-1} = IA^{-1} = I\).

Similarly \((B^{-1}A^{-1})(AB) = I\).

Thus \( AB \) is invertible and \((AB)^{-1} = B^{-1}A^{-1}\).

Notational conventions. Because matrix multiplication is not commutative, write \( A^{-1} \) and NOT \( I/A \) or \( I \). Some computer packages, however, violate the conventions. In Matlab,

\[\text{X} = A\backslash B\] is a solution to \( A\star X = B \)

\[\text{X} = A/B\] is a solution to \( X\star A = B \)

We strongly urge you not to adopt notations like this outside Matlab.

**THEOREM.** (ON Thm 8.21.) Let \( A \) be invertible. Then \((A^{-1})^{-1} = A\).

If also \( k \neq 0 \) then \((kA)^{-1} = \frac{1}{k}A^{-1}\).

**Definition.** Let \( A = (a_{ij})_{n\times n} \). Define the transpose of \( A \), denoted \( A^T \), by \( A^T = (a_{ji})_{n\times m} \), i.e. interchange rows and columns.

We shall think of the elements of \( \mathbb{R}^n \) as column vectors, and denote the set of \( n \times n \) real matrices by \( \mathbb{R}^{n\times n} \). Given \( \mathbf{v} \in \mathbb{R}^n \), we denote the \( i \)-th component of \( \mathbf{v} \) by \( v_i \) or \( (\mathbf{v})_i \); e.g., the \( i \)-th component of the \( k \)-th standard basis vector \( e_k \) is

\[
(e_k)_i = \delta_{ki} = \begin{cases} 
1 & \text{if } i = k, \\
0 & \text{if } i \neq k.
\end{cases}
\]
The transpose of \( v \) is denoted by \( v^T \), and we remark that

\[
v_k = e_k^T v = v^T e_k.
\]

Given \( A \in \mathbb{R}^{n \times n} \), we denote the \( i, j \) entry of \( A \) by \( a_{ij} \) or \( (A)_{ij} \), and write \( A = [a_{ij}] \). The simple properties

\[
e_i^T A = \text{row } i \text{ of } A, \quad Ae_j = \text{column } j \text{ of } A, \quad e_i^T Ae_j = a_{ij},
\]

will prove to be very useful. **THEOREM.** (ON Thm 8.4(3).) \((AB)^T = B^T A^T\).

There is a nice picture of this in [Str] §2.7 2nd ed. p99, 3rd ed. p109.

**THEOREM.** (ON Thm 8.21.) \((A^T)^{-1} = (A^{-1})^T\).

This subsection has collected some theoretical results concerning matrix inverses. In some treatments of linear algebra, including that in [ON], these matters are treated after the algorithmic details of how one calculates for solving linear systems and for finding inverses. The next subsections of these notes briefly revise your first year work ([UWA1] Chpt 1) on these algorithmic details.

**Elementary row operations and elementary matrices (ON §8.3)**

**Definition.** (ON Defn 8.6.) Let \( A \) be an \( n \) by \( m \) matrix. We define three types of elementary row operations, (e.r.o.s.)

1. interchanging two rows, or
2. multiplying a row of \( A \) by a nonzero constant, or
3. adding a multiple of one row to another row.

**Definition.** (ON Defn 8.7.) An elementary matrix is a matrix formed by performing an e.r.o. on an identity matrix \( I_n \).

**THEOREM.** (ON Thm 8.6.) Let \( A \) be an \( n \) by \( m \) matrix. Suppose \( B \) is formed from \( A \) by an e.r.o.. Let \( E \) by the elementary matrix formed by performing the same e.r.o. on \( I_n \). Then \( B = EA \).

**EXAMPLES.**

\[
E_{12}(c) = \begin{bmatrix}
1 & 0 & 0 \\
c & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

Multiplying out \( E_{12}(c)A \) produces a matrix whose 1st and 3rd rows are unchanged, while the 2nd row is the sum of the original 2nd row plus \( c \) times the 1st row.

\[
P_{12} = \begin{bmatrix}
0 & 1 & 0 \\
1 & 0 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

Multiplying out \( P_{12}A \) swaps the 1st and 2nd rows of \( A \).

**A generalization.**

**Definition.** A matrix of the form

\[
E(u, v, c) = I - cuv^T,
\]

where \( u, v \) are column vectors and \( c \) is a scalar, is called a generalized elementary matrix, which we abbreviate to g.e.m..

It is easy to check that

\[
E(u, v, c_1)E(u, v, c_2) = E(u, v, c_1 + c_2 - c_1c_2v^Tu).
\]
It follows that when
\[ c_1^{-1} + c_2^{-1} = v^T u, \]
\[ E(u, v, c_1) = (E(u, v, c_2))^{-1}. \]

Next we represent our ordinary elementary matrices as g.e.m.s.

1. Interchanging rows \( i \) and \( j \) is achieved by pre-multiplying by \( P_{i,j} \) where
\[ P_{i,j} = E(e_i - e_j, e_i - e_j, 1). \]

2. Multiplying the \( i \)-th row of by a nonzero constant \( c \),
\[ M_i(c) = E(e_i, e_i, 1 - c). \]

3. Adding \( c \) times row \( k \) to row \( i \),
\[ A_{i,k}(c) = E(e_i, e_k, -c). \]

A class of g.e.m.s of use in describing LU-factorization are the lower-triangular g.e.m.s of the form
\[ E(m, e_k, 1) \quad \text{where } e_i^T m = 0 \quad (i = 1, 2, \ldots, k). \]

Row equivalence

**Definition.** (ON Defn 8.8.) If \( A \) and \( B \) are matrices of the same size, we say that \( A \) is row equivalent to \( B \) if \( B \) can be obtained through a sequence of matrices \( A_0 = A, A_{i+1} \) found from \( A_i \) by an elementary row operation.

**THEOREM.** (ON Thm 8.7.)

1. (Reflexive Property.) Every matrix is row equivalent to itself.
2. (Symmetric Property.) If \( A \) is row equivalent to \( B \), then \( B \) is row equivalent to \( A \).
3. (Transitive Property.) If \( A \) is row equivalent to \( B \), and \( B \) is row equivalent to \( C \), then \( A \) is row equivalent to \( C \).

Gauss elimination and rref (ON §8.4)

**Definitions.** A matrix \( M \) is said to be an echelon matrix, or a matrix in row-echelon form, e.g.
\[ M = \begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & 0 & * \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}, \]

if
(i) the zero rows, if any, are at the bottom, and
(ii)If \( R_1, R_2, \ldots, R_i \) are the nonzero rows (in order from the top) the leading entry (i.e. first nonzero entry) of \( R_{i+1} \) is strictly further right than the leading entry of \( R_i \).

A result from early in your first year course ([UWA1] §1.2 Theorem 1.1) is the following.

**THEOREM.** Every matrix is row equivalent to an echelon matrix.

Gauss elimination, treated in your first year course, is a process of reaching an echelon matrix by using e.r.o.s.

ON p341, [Str] define reduced row echelon form, for a matrix which is an echelon matrix and also (iii) where the first nonzero (leading entry) in any nonzero row \( R_i \) is a 1,
(iv) if any row has its leading entry in column \( k \), all other entries in column \( k \) are zero.

[ON] has an unconventional abbreviation of this to reduced matrix, which we recommend that you do not use. Others, more conventionally, use the first letters of the four words, \( \text{rref} \).

In real float numeric calculations, the rref form of a matrix is not used as much as the earlier Gauss elimination (which we will discuss as \( LU \)-factorization in a later section). There are, however, theoretical uses of rref and later in these notes we will also illustrate its use in hand calculations of inverses of relatively small matrices.

**THEOREM.** (ON Thm 8.8.) Every matrix is row equivalent to a matrix in reduced row echelon form.

**THEOREM.** (ON Thm 8.9.) Let \( A \) be a matrix. Then there is exactly one matrix in reduced row echelon form that is row equivalent to \( A \). (We denote this matrix \( \text{rref}(A) \).)

**THEOREM.** (ON Thm 8.10.) Let \( A \) be a matrix. Then there is a square matrix \( \Omega \) such that \( \Omega A = \text{rref}(A) \).

We will now have a break from theoretical considerations, and uses of rref in the theory, and do some (first year) sums using e.r.o.s and rref.

**e.r.o.s to solve** \( Ax = b \)

One solves \( Ax = b \) by transforming it to a simpler, echelon, system. The process is usually done in two steps

1. Gauss elimination,
2. back substitution.

The back substitution part becomes easier if we do more work - Gauss-Jordan reduction - instead of just Gauss elimination at step 1. (Wilhelm Jordan was an engineer. The large systems of linear equations considered both by him and by Gauss came from surveying, geodesy applications.)

Gauss-Jordan elimination is the process of forming \( \text{rref}([A:b]) \). There are examples of Gauss-Jordan elimination in AEM text books, e.g. [ON] Example 8.29.

**rref, e.r.o.s to find** \( A^{-1} \) ([ON] §8.8)

This topic was treated in your first year course. See [UWA1] §1.7.

Given square matrix \( A \). We can use e.r.o.s to find \( A^{-1} \) if it exists.

**THEOREM.** (ON Thm 8.22.) If an \( n \) by \( n \) matrix \( A \) can be transformed into the \( n \) by \( n \) identity matrix \( I_n \) by a sequence of elementary row operations, then \( A \) is nonsingular. The same sequence of operations that transforms \( A \) into the identity \( I_n \) will also transform \( I_n \) into \( A^{-1} \). In symbols

\[
[A:I] \sim [I:A^{-1}].
\]

Let \( A \) be an arbitrary nonsingular matrix. Let \( \theta_1, \theta_2 \ldots \theta_s \) be any sequence of e.r.o.s that transforms \( A \) into \( I \). Then the same sequence of e.r.o.s transforms \( I \) into \( A^{-1} \).

Here is an example. (For others, see your first year notes, ON Ex 8.33, 8.34 pp375-376.)

**EXAMPLE.** Find \( A^{-1} \) if

\[
A = \begin{bmatrix}
-1 & 2 & 1 \\
0 & 1 & -2 \\
1 & 4 & -1 \\
\end{bmatrix}.
\]

Solution.

\[
\begin{bmatrix}
A & I \\
\end{bmatrix} = \begin{bmatrix}
-1 & 2 & 1 & : & 1 & 0 & 0 \\
0 & 1 & -2 & : & 0 & 1 & 0 \\
1 & 4 & -1 & : & 0 & 0 & 1 \\
\end{bmatrix} \sim \begin{bmatrix}
1 & -2 & -1 & : & -1 & 0 & 0 \\
0 & 1 & -2 & : & 0 & 1 & 0 \\
0 & 6 & -2 & : & 1 & 0 & 1 \\
\end{bmatrix} - R_1
\]

\[
R_1 + R_3
\]

13
Divide the 3rd row of the last matrix by 6, and continue:

\[
\begin{bmatrix}
A & I
\end{bmatrix} \sim \begin{bmatrix}
1 & -2 & -1 & : & -1 & 0 & 0 \\
0 & 1 & 0 & : & -\frac{1}{6} & 0 & \frac{1}{6}
0 & 1 & -2 & : & 0 & 1 & 0
\end{bmatrix} R_2 \leftrightarrow R_3
\]

\[
\sim \begin{bmatrix}
1 & 0 & -1 & : & -\frac{2}{3} & 0 & \frac{1}{3}
0 & 1 & 0 & : & \frac{1}{3} & 0 & \frac{1}{3}
0 & 0 & +1 & : & +\frac{1}{12} & -\frac{1}{2} & +\frac{1}{12}
\end{bmatrix}
\]

\[
R_1 + 2R_2 \quad -\frac{1}{2}(R_3 - R_2)
\]

\[
\sim \begin{bmatrix}
1 & 0 & 0 & : & -\frac{7}{12} & -\frac{1}{2} & \frac{5}{12}
0 & 1 & 0 & : & \frac{1}{6} & 0 & \frac{1}{6}
0 & 0 & 1 & : & \frac{1}{12} & -\frac{1}{2} & \frac{1}{12}
\end{bmatrix} R_1 + R_3
\]

Our calculation gives

\[A^{-1} = \frac{1}{12} \begin{bmatrix}
-7 & -6 & 5 \\
2 & 0 & 2 \\
1 & -6 & 1
\end{bmatrix}.\]

Check by multiplying the matrices:

\[AA^{-1} = \frac{1}{12} \begin{bmatrix}
7 + 4 + 1 & 6 - 6 & -5 + 4 + 1 \\
2 - 2 & 12 & 2 - 2 \\
-7 + 8 - 1 & -6 + 6 & 5 + 8 - 1
\end{bmatrix}.\]

Similarly (an exercise for you), multiply out \(A^{-1}A\) and verify that it also is the identity matrix.

EXAMPLE. Find \(A^{-1}\) if

\[A = \begin{bmatrix}
4 & 2 & 3 \\
2 & 1 & 0 \\
-1 & -2 & 0
\end{bmatrix}.
\]

Solution.

\[
\begin{bmatrix}
A & I
\end{bmatrix} = \begin{bmatrix}
4 & 2 & 3 & : & 1 & 0 & 0 \\
2 & 1 & 0 & : & 0 & 1 & 0 \\
-1 & -2 & 0 & : & 0 & 0 & 1
\end{bmatrix} \sim \begin{bmatrix}
4 & 2 & 3 & : & 1 & 0 & 0 \\
0 & -\frac{3}{2} & \frac{3}{4} & : & -\frac{1}{2} & 0 & 1 \\
0 & 0 & -\frac{3}{2} & : & -\frac{1}{2} & 1 & 0
\end{bmatrix} R_2 - \frac{1}{2}R_1 R_3 + \frac{1}{2}R_1
\]

Continuing,

\[
\begin{bmatrix}
A & I
\end{bmatrix} \sim \begin{bmatrix}
4 & 2 & 3 & : & 1 & 0 & 0 \\
0 & -\frac{3}{2} & \frac{3}{4} & : & \frac{1}{4} & 0 & 1 \\
0 & 0 & -\frac{3}{2} & : & -\frac{1}{2} & 1 & 0
\end{bmatrix} R_2 \leftrightarrow R_3
\]

\[
\sim \begin{bmatrix}
1 & \frac{1}{2} & \frac{3}{4} & : & \frac{1}{4} & 0 & 0 \\
0 & 1 & -\frac{1}{2} & : & -\frac{1}{6} & 0 & -\frac{2}{3}
0 & 0 & 1 & : & \frac{1}{3} & -\frac{4}{3} & 0
\end{bmatrix} \frac{1}{4}R_1 - \frac{1}{2}R_2 - \frac{1}{3}R_3
\]

\[
\sim \begin{bmatrix}
1 & 0 & 0 & : & 0 & \frac{2}{3} & \frac{1}{3} \\
0 & 1 & 0 & : & 0 & -\frac{1}{3} & -\frac{2}{3}
0 & 0 & 1 & : & \frac{1}{3} & -\frac{2}{3} & 0
\end{bmatrix} R_1 - \frac{1}{6}R_2 - \frac{1}{3}R_3
\]

\[R_2 + \frac{1}{2}R_3
\]
Our calculation gives
\[ A^{-1} = \frac{1}{3} \begin{bmatrix} 0 & 2 & 1 \\ 0 & -1 & -2 \\ 1 & -2 & 0 \end{bmatrix} . \]

Check by multiplying the matrices.

**Differential equations uses of matrix inverses**

**The variation of parameters formula**

In an earlier handout we reviewed what you had learnt about solving a single first-order linear d.e. for a single function \( y(t) \). We treated the d.e.

\[ \frac{dy}{dt} - A(t)y = f(t) \]  

(N1)

and also used the homogeneous equation, which we will now call \((H_1)\). \((H_1)\) is just the above with \( f(t) \) replaced by zero. We found the formula – pretty much the same formula as you have seen in first year –, that with \( \phi(t) \) the solution of \((H_1)\), the solution of \((N_1)\) is

\[ y(t) = \phi(t) \left( y(0) + \int_0^t \phi(s)^{-1} f(s) ds \right) . \]

Now consider systems of first order equations, e.g.

\[ \begin{align*}
\frac{dy_1}{dt} - a_{11}(t)y_1 - a_{12}y_2 &= f_1(t) \\
\frac{dy_2}{dt} - a_{12}(t)y_1 - a_{22}y_2 &= f_2(t)
\end{align*} \]

Of course we can write this (and similar larger systems)

\[ \frac{d\mathbf{y}}{dt} - A(t)\mathbf{y} = \mathbf{f} . \]  

(N)

Here \( \mathbf{y} \) is a (column) vector of functions – it is what we have to find; 
\( \mathbf{f} \) is a column vector of given functions – the forcing; 
\( A \) is a square matrix – its entries \( a_{ij}(t) \) are given functions.

We have already been happily differentiating vectors. Suppose we have some matrix, which we will call \( \Phi \) later, and its entries \( \phi_{ij}(t) \) are functions of \( t \). Again we can make sense of \( \frac{d\Phi}{dt} \) as the matrix whose \( i, j \)-entry is \( \frac{d\phi_{ij}}{dt} \). Now suppose we somehow have a nonsingular solution of

\[ \frac{d\Phi}{dt} - A(t)\Phi = 0 , \]

where the 0 at the right is a square zero matrix. Actually if we have one of these, by defining \( \Phi_1(t) = \Phi(t)\Phi(0)^{-1} \) (i.e. multiplying at the right by an inverse), we have that

\[ \frac{d\Phi_1}{dt} - A(t)\Phi_1 = 0 , \quad \Phi_1(0) = I . \]  

(H)

The ‘variation of parameters’ formula solving the nonhomogenous system \((N)\) is:

\[ \mathbf{y}(t) = \Phi_1(t) \left( \mathbf{y}(0) + \int_0^t \Phi_1(s)^{-1} \mathbf{f}(s) ds \right) , \]

where the \( \Phi_1(s)^{-1} \) is a matrix inverse. It is actually fairly easy to prove this. One ingredient is that for matrices \( M_1 \) and \( M_2 \), we have

\[ \frac{d}{dt} (M_1 M_2) = M_1 \frac{dM_2}{dt} + \frac{dM_1}{dt} M_2 . \]
We have to be a bit careful as a consequence of matrix multiplication being non-commutative. The proof of the formula is not in the exam!! The important thing is that there is a formula, and it underpins how your CAS would go about seeking formulae for solutions of linear systems. Needless to say, humans don’t do this sort of sum by hand any more. It is truly hideous to do all the integrals, and rather depressing as mostly – by machine, or by hand – they can’t be done exactly... Life gets a lot easier when \( A(t) \) is constant in time. Also, there are important engineering applications of the constant coefficient case, and we will return to it after learning about the linear algebra needed to do the sums, namely about eigenvalues.

**rank and related topics (ON §8.5)**

((Str) §3.2-3.3.)

Consider the problem \( Ax = b \). An earlier section on matrix inverses gives some theory for how many solutions to expect when \( A \) is square and invertible, i.e. one solution. We now introduce some more definitions which are useful not just when \( A \) is square.

Here is the definition of rank in the way you had it in your first year course.

**Definition.** The *row rank* of a matrix \( A \) is the number of linearly independent rows in \( A \). The *column rank* of the matrix \( A \) is the number of linearly independent columns in \( A \). ([Str] §3.1 2nd ed. p105, 3rd ed. p114.)

**Definition.** (ON Defn 8.11.) The *row space* of a matrix \( A \) is span of rows in \( A \). The *column space* of a matrix \( A \) is span of columns in \( A \).

The proof of the following memorable result is not required in this course.

**THEOREM.** (ON Thm 8.13.) If \( A \) is a matrix, then the row and the column spaces of \( A \) have the same dimension. In other words

\[
\text{rowrank}(A) = \text{columnrank}(A).
\]

Henceforth call them rank of \( A \), denoted \( \text{rank}(A) \). (See also [UWA1] §3.5 p109, [HDR] Chpt 3 p121.)

A result stated in ON Thm 8.9 can now be restated as follows.

**THEOREM.** Matrices that are row equivalent have the same rank.

In fact, [ON] uses this as the starting point of his definition of rank, which is, in the end, completely equivalent to the definition you had at first year.

Here are some results concerning square matrices.

**THEOREM.** (ON Thm 8.11.) Let \( A \) be an \( n \) by \( n \) matrix. Then \( \text{rank}(A) = n \) if and only if \( \text{rref}(A) = I_n \).

**THEOREM.** (ON Thm 8.22.) If \( A \) is \( n \) by \( n \) then \( A \) nonsingular \( \iff \text{rank}(A) = n \).

**Some rank inequalities**

\[
\text{rank}(A + B) \leq \text{rank}(A) + \text{rank}(B).
\]

\[
\text{rank}(AB) \leq \max(\text{rank}(A), \text{rank}(B)).
\]

If \( P \) and \( Q \) are invertible matrices

\[
\text{rank}(PAQ) = \text{rank}(A).
\]

\[
\text{rank}(AB) \geq \text{rank}(A) + \text{rank}(B) - (\text{number of rows of } A).
\]
Definition. Let $A$ be a matrix. The nullspace (or kernel) of $A$ is defined by
\[
\text{nullspace}(A) = \{ w | Aw = 0 \},
\]
and its dimension is sometimes written nullity$(A)$.

Another major result from your first year course is the following. (See [UWA1] §3.5 p110.)

**Rank-nullity THEOREM.** (ON Thm 8.15.) Let $A$ be a matrix.
\[
\text{rank}(A) + \dim(\text{nullspace}(A)) = \text{number of columns of } A.
\]

We now consider square matrices again.

**THEOREM.** (ON Thm 8.23.) $A$ is $n \times n$. Then
\[
A \text{ is singular } \iff \exists x \neq 0 \text{ such that } Ax = 0.
\]

**Systems of linear equations (ON §8.7)**

**Definition.** A system of linear equations $Ax = b$ is said to be *homogeneous* if $b = 0$ and *nonhomogeneous* otherwise.

Suppose that $Ax_p = b$. Then, every vector of the form
\[
x_p + w \quad \text{with} \quad w \in \text{nullspace}(A)
\]
solves $Ax = b$ and any $x$ solving $Ax = b$ is in $x_p + \text{nullspace}(A)$. See Figure 3.3 of [Str].

The preceding paragraph gives us how many solutions there are if there are any. The question of whether there are any is the next topic.

The three possibilities for solutions of systems $Ax = b$ are as follows. (See ON §8.7, [Str] §2.1 or HDR Example 3.12 p118.)

![Figure 6: The three cases for linear systems: no solutions (two parallel distinct lines); a unique solution (two nonparallel lines); infinitely many solutions (two parallel coincident lines).]

**Unique solution**
\[
e.g. \quad x_1 + x_2 = 2 \\
x_1 - x_2 = 0
\]

**Inconsistent, or no solution**
\[
e.g. \quad x_1 + x_2 = 2 \\
x_1 + x_2 = 0
\]

**Infinitely many solutions**
\[
e.g. \quad x_1 + x_2 = 2 \\
2x_1 + 2x_2 = 4
\]
In this last case the solutions are $x_1 = t$, $x_2 = 2 - t$ for all $t$.

The preceding exemplifies the results (HDR Thm 3.7).

<table>
<thead>
<tr>
<th>Number of solutions of $Ax = b$</th>
<th>rank($([A:b])$) and rank($A$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>rank($([A:b])$) ≠ rank($A$)</td>
</tr>
<tr>
<td>precisely 1</td>
<td>rank($([A:b])$) = rank($A$) = number of columns of $A$</td>
</tr>
<tr>
<td>more than 1</td>
<td>rank($([A:b])$) = rank($A$) &lt; number of columns of $A$</td>
</tr>
</tbody>
</table>

Suppose that the system $Ax = b$ is consistent and that $x_p$ is one particular solution. Then the solution set of $Ax = b$ is $\{x_p + z | z \in W\}$ where $W = \{z | Az = 0\}$.

EXAMPLE/APPLICATION. The three points $(x_i, y_i)$, $i = 1, 2, 3$ in the plane are collinear $\Leftrightarrow$ the rank of the following matrix is less than 3:

\[
\begin{bmatrix}
x_1 & y_1 & 1 \\
x_2 & y_2 & 1 \\
x_3 & y_3 & 1
\end{bmatrix}
\]

**THEOREM.** ([Str] 3A.) The system $Ax = b$ is solvable if and only if $b$ is in the column space of $A$.

A fundamental theorem

**LEMMA.** If $w$ is in the nullspace of $A^T$ and $b$ is in the column space of $A$, then $w^Tb = 0$.

**Proof.** ([Str] §4.1 Thm 4B.) Let $b = Ax$.

\[
w^Tb = w^T(Ax) = (w^T A)x = (A^T w)^T x = 0^T x = 0,
\]

as required.

A better result is the fundamental theorem which follows. First, however, a definition and another result.

**Definition.** Define the set $U^\perp$ from a given vector subspace $U$ of $\mathbb{R}^m$ by

\[
U^\perp = \{b | u^T b = 0 \forall u \in U\}.
\]

It turns out that $U^\perp$ is a vector subspace of $\mathbb{R}^m$, that

\[
U \cap U^\perp = \{0\},
\]

and

\[
\text{dimension}(U) + \text{dimension}(U^\perp) = m.
\]

We say that $U^\perp$ is the orthogonal complement of $U$.

**Remark:** This definition makes sense in a more general setting: see the later section on inner product spaces.

**THEOREM.** ([Str] §4.1.)

\[
Ax = b \text{ has a solution} \\
\Leftrightarrow b \in \text{columnspace}(A) \\
\Leftrightarrow w^T b = 0 \forall w \in \text{nullspace}(A^T).
\]
Proof. The result $\Rightarrow$ is easy: it is the result we have in the Lemma above.

The result $\Leftarrow$ is a bit more difficult. One proof can be devised using the dimension results we have earlier: the fact that the dimensions of row space and column space of any matrix is the same, and the rank-nullity theorem. Supposing $A$ is $m$ by $n$, rank $r$, these dimensions can be summarised in the following table:

$$
\begin{align*}
\dim(\text{column space}(A)) &= r \\
\dim(\text{null space}(A)) &= n - r
\end{align*}
$$

We also have that 

$$
\text{null space}(A^T) \cap \text{column space}(A) = \{0\}.
$$

(For suppose $b$ is in both the subspaces at left. Let $b = Ax$. Then 

$$b^T b = b^T A x = 0,$$

so $b = 0$.) From these we have

$$
\text{null space}(A^T) \oplus \text{column space}(A) = \mathbb{R}^n
$$

Next we consider $\text{null space}(A^T)^\perp$. The discussion after the definition of $U^\perp$ shows that

$$
\dim(\text{null space}(A^T)^\perp) = \dim(\text{column space}(A)).
$$

By careful consideration, we get, not only that the dimensions are equal, but

$$
\text{null space}(A^T)^\perp = \text{column space}(A).
$$

This gives us that the existence of solutions can be established when the rhs $b$ is given to be in $(\text{null space}(A^T))^\perp$, as required.

Allowing ourselves to re-state results given above, we have:

**Fundamental Theorem of Linear Algebra.** (([Str] §4.1 2nd ed. p168, 3rd ed. 187.) Let $A$ be an $m$ by $n$ matrix.

The null space (of $A$) is the orthogonal complement of the row space (in $\mathbb{R}^n$).

The left null space is the orthogonal complement of the column-space (in $\mathbb{R}^m$).

Having said all this, in practice engineers are likely to use software which automates linear equation solving. The fundamental theorem underpins some calculations. There is, for example, a generalization of the method of undetermined coefficients for constant coefficient linear systems of d.e.s, and when the forcing term solves the homogeneous system one way of organizing the calculations involves using the fundamental theorem. Fundamental as the Fundamental Theorem is, we don’t want to waste your time on messy sums.

See the diagram, Strang 2nd ed. p169, 3rd ed. p188:
More on matrix inverses (ON §8.8)

Here are some useful facts from before, collected into one theorem.

**THEOREM.** The following conditions are equivalent for \( n \times n \) matrix \( A \).

1. \( A \) is invertible (= nonsingular).
2. If \( y^T A = 0 \) where \( y \) is \( n \times 1 \), then \( y = 0 \).
3. \( \text{rank}(A) = n \).
4. \( A \) can be transformed to the \( n \times n \) identity matrix \( I_n \) by e.r.o.s.
5. \( A \) is a product of elementary matrices.

We can halve the amount of work in checking our calculation of a matrix inverse with the following result that, for square matrices, left-invertibility implies invertibility.

**THEOREM.** Let \( A \) and \( B \) be \( n \times n \) matrices. If \( AB = I_n \), then also \( BA = I_n \) so \( A \) and \( B \) are invertible and \( A = B^{-1} \) and \( B = A^{-1} \).

**Proof.** It suffices to show that \( A \) is invertible. (Then left-multiplying \( AB = I \) by \( A^{-1} \) gives \( B = A^{-1} \).) Use property 2 of the preceding theorem.

\[
y^T A = 0 \quad \Rightarrow \quad 0 = (y^T A) B = y^T (AB) = y^T I = y^T,
\]

\[
\Rightarrow \quad y = 0,
\]

so \( A \) is invertible.
 BLOCK PARTITIONED MATRICES

We have already seen that it can be useful to augment matrices, i.e. starting with matrices $A_1$ and $A_2$ with the same number of rows, to build another matrix $A$ with the same number of rows by ‘putting them side by side’ $A = [A_1 : A_2]$. Obviously we could do the same with any number of matrices $A_1, \ldots, A_l$. Clearly there are all sorts of results related to this process, e.g. $\operatorname{rank}(A) \geq \min(\operatorname{rank}(A_j))$.

As elsewhere, we can do the same process using rows rather than columns. We then stack matrices, which now have the same number of columns, one on top of the other.

This general idea of building up big matrices from lots of small ones can be extended. Suppose $A$ is a big matrix, $m$ by $n$, and

$$m = \sum_{i=1}^{k} m_i, \quad n = \sum_{j=1}^{l} n_j.$$  

Then one can partition $A$ into several (say $k$ by $l$) little matrices $X_{ij}$ of size $m_i$ by $n_j$, (each $X_{ij}$ being called a block) by ‘drawing dotted lines’ after each row numbered $\sum_{i=1}^{k'} m_i, k' = 1, \ldots, k$, and after each column numbered $\sum_{j=1}^{l'} n_j, l' = 1, \ldots, l$.

Here is the picture

$$A = \begin{bmatrix}
X_{11} & : & X_{12} & : & \cdots & : & X_{1l} \\
\vdots & : & \vdots & : & \ddots & : & \vdots \\
X_{21} & : & X_{22} & : & \cdots & : & X_{2l} \\
\vdots & : & \vdots & : & \ddots & : & \vdots \\
\vdots & : & \vdots & : & \ddots & : & \vdots \\
X_{k1} & : & X_{k2} & : & \cdots & : & X_{kl}
\end{bmatrix}.$$

(Numerical methods for PDEs, not treated in the 2nd year Engineering Maths courses, are one area where very large block partitioned matrices arise in engineering applications. There is some discussion in ON §16.13. There are many other different applications, e.g. in the truss problem mentioned earlier.

[HDR] p145 Prob P3.38-P3.39 treats block partitioned matrices. [ON] §8.9 Q18 gives the definition of block diagonal, i.e. block partitioned as above with, additionally, $X_{ij}$ for $i \neq j$ all zero matrices, and all the $X_{ii}$ square.

Good elementary linear algebra texts contain the definition of block partitioned and special cases like block diagonal. See, for example [Str] §2.3 pp57-65.

A special case which is easy to describe is if all the $m_i$ are equal, say, are $\hat{m}$, and all the $n_i$ are equal, say, are $\hat{n}$. Then $A$ can be identified with a $k$ by $l$ array of $(\hat{m} \times \hat{n})$ small arrays.

A special case which is easy to describe is if all the $m_i$ are equal, say, are $\hat{m}$, and all the $n_i$ are equal, say, are $\hat{n}$. Then $A$ can be identified with a $k$ by $l$ array of $(\hat{m} \times \hat{n})$ small arrays.

A special case which is easy to describe is if all the $m_i$ are equal, say, are $\hat{m}$, and all the $n_i$ are equal, say, are $\hat{n}$. Then $A$ can be identified with a $k$ by $l$ array of $(\hat{m} \times \hat{n})$ small arrays.

A special case which is easy to describe is if all the $m_i$ are equal, say, are $\hat{m}$, and all the $n_i$ are equal, say, are $\hat{n}$. Then $A$ can be identified with a $k$ by $l$ array of $(\hat{m} \times \hat{n})$ small arrays.

type littlearray = array[1 .. $\hat{m}$, 1 .. $\hat{n}$] of scalar

A: array[1 .. k, 1 .. l] of littlearray

More generally, when the $m_i$ and/or $n_j$ are different, the data structure is more elaborate.

Partitioned matrices can sometimes be added and multiplied, in ways where the partitioning is useful. Again, good linear algebra texts explain this.
Some special sorts of matrices

Definition. A square matrix in which all the entries off the main diagonal are zero is called a diagonal matrix.

Definition. (ON p386.) A square matrix is said to be lower triangular if all entries above and right of the main diagonal are zero.

Later, in connection with numerical solution of linear systems by LU factorisation we will prove facts like the following.

THEOREM. The product of two n by n lower triangular matrices is lower triangular.

Definition. A lower triangular matrix is said to be unit lower triangular if its diagonal entries are all one.

THEOREM. The product of two n by n unit lower triangular matrices is unit lower triangular. The inverse of a unit lower triangular matrix is unit lower triangular.

There are corresponding definitions and results for upper triangular matrices.

Definition. (ON p422.) A square matrix is said to be symmetric if \( A = A^T \).

The importance of symmetric matrices will be seen in later chapters.

Definition. (ON p389.) A square matrix is said to be skew-symmetric if \( A = -A^T \).

Skew-symmetric matrices, although interesting, are less important in practice than are symmetric ones.

A generalisation of ‘diagonal’ matrices is the following.

Definition. A \( n \) by \( n \) matrix is said to be banded if \( A = (a_{ij}) \) satisfies

\[
|i - j| > k \implies a_{ij} = 0,
\]

and then its bandwidth is defined to be \( 2k + 1 \).

Thus a diagonal matrix has bandwidth 1. If the bandwidth is 3 the matrix is said to be tridiagonal. We have already banded matrices in the example involving the stresses in a truss (bridge) with given loads.

An \( n \) by \( n \) matrix with each entry zero except for precisely one 1 in each row and in each column is called a permutation matrix. E.g.

\[
\begin{bmatrix}
0 & 1 \\
1 & 0
\end{bmatrix},
\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
1 & 0 & 0
\end{bmatrix}.
\]

A square real matrix \( Q \) such that \( QQ^T = I \) is said to be orthogonal. (ON p419)

A matrix \( P \) such that \( P^2 = P \) is said to be idempotent, or if \( P^k = P \) idempotent of order \( k \).

A symmetric idempotent matrix is called a projection.

There are several other special sorts of matrices, projection, positive definite (ON p428), hermitian (ON p432), unitary (ON p430), normal. We will introduce definitions later as we need them.

References


