OVERVIEW OF MATRIX ALGEBRA

(or All you ever wanted to know about Linear Algebra but were afraid to ask...)

A matrix is simply a rectangular array of "elements" arranged in a series of \( m \) rows and \( n \) columns. The order of a matrix is the specification of the number of rows by the number of columns. Elements of a matrix are given as \( a_{ij} \), where the value of \( i \) specifies the row position and the value of \( j \) specifies the column position; so \( a_{ij} \) identifies the element at position \( i,j \).

An element can be a number (real or complex), algebraic expression, or (with some restrictions) a matrix or matrix expression. For example:

\[
A = \begin{bmatrix}
12 & 4 & 10 \\
8 & 1 & 11 \\
15 & 3 & 11 \\
14 & 1 & 11 \\
\end{bmatrix} = \begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33} \\
a_{41} & a_{42} & a_{43} \\
\end{bmatrix}
\]

This matrix, \( A \), has order \( 4 \times 3 \), the element \( a_{23} = 11 \), \( a_{13} = 10 \), etc. The notation for matrices is not always consistent but is usually one of the following schemes:

- **matrix** - designated by a bold letter (most common); capital letter; or letter with under-score, brackets or hat (\(^\wedge\)). Order is also sometimes given: \( A_{(4,3)} = A \) is \( 4 \times 3 \).
- **order** - always given as row \( x \) column but uses letters \( n,m,p \) differently: \( n \text{(row)} \times m \text{(column)} \) or \( m \text{(row)} \times n \text{(column)} \).
- **element** - most commonly \( a_{ij} \) with \( i = \text{row}; j = \text{column} \); (sometimes \( k,l,p \))

Advantages of matrix algebra mainly lies in the fact that it provides a concise and simple method for manipulating large sets of numbers or computations, making it ideal for computers. Also (1) Compact form of matrices allows convenient notation for describing large tables of data; (2) Operations allow complex relationships to be seen which would otherwise be obscured by the sheer size of the data (i.e., it aids clarification); and (3) Most matrix manipulation involves just a few standard operations for which standard subroutines are readily available.

As a convention with data matrices (i.e., the elements represent data values), the columns usually represent the different variables (e.g., one column contains temperatures, another salinity, etc.) while rows contain the samples (e.g., the values of the variable at each depth or time). Since there are usually more samples than variables, such data matrices are usually rectangular having more rows \( (m) \) than columns \( (n) \) - order \( m \times n \) where \( m > n \).

**Column vector** is a matrix containing only a single column of elements:
Row vector is a matrix containing only a single row of elements:

\[ a = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \]

Size of a vector is simply the number of elements it contain (= \( n \), in both examples above). Null matrix, written as \( 0 \) or \( 0_{(m,n)} \) has all elements equal to 0 - it plays the role of zero in matrix algebra.

Square matrix has the same numbers of rows as columns, so order is \( n \times n \).

Diagonal matrix is a square matrix with zero in all positions except along the principal (or leading) diagonal:

\[ D = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 6 \end{bmatrix} \]

or

\[ d_{ij} = \begin{cases} 0 & \text{for } i \neq j \\ \text{non-zero} & \text{for } i = j \end{cases} \]

This type of matrix is important for scaling rows or columns of other matrices. The Identity matrix \( (I) \) is a diagonal matrix with all of the nonzero elements equal to one. Written as \( I \) or \( I_n \); it plays the role of 1 in matrix algebra \( (A \cdot I = I \cdot A = A) \). A Lower triangular matrix \( (L) \) is a square matrix with all elements equal to zero above the principal diagonal:

\[ L = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 7 & 0 \\ 8 & 2 & 6 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 8 \end{bmatrix} \]

or

\[ l_{ij} = \begin{cases} 0 & \text{for } i < j \\ \text{non-zero} & \text{for } i \geq j \end{cases} \]

An Upper triangular matrix is a square matrix with all elements equal to zero below the principal diagonal

\[ U = \begin{bmatrix} 0 & 0 \\ 3 & 7 \\ 8 & 2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 3 & 7 \end{bmatrix} \]

or

\[ u_{ij} = \begin{cases} 0 & \text{for } i < j \\ \text{non-zero} & \text{for } i \leq j \end{cases} \]

(If one multiplies 2 triangular matrices of the same form, the result is a third matrix of the same form).

We also have Fully populated matrix which is a matrix with all of its elements nonzero, Sparse matrix which is a matrix with only a small proportion of its elements nonzero, and Scalar which simply is a number (i.e., matrix with a single element).
Matrix transpose (or transpose of a matrix) is obtained by interchanging the rows and columns of a matrix. So row $i$ becomes column $i$ and column $j$ becomes row $j$ (also the order of the matrix is reversed).

\[
A = \begin{bmatrix}
1 & 14 \\
6 & 7 \\
8 & 2
\end{bmatrix}; A^T = \begin{bmatrix}
1 & 6 & 8 \\
14 & 7 & 2
\end{bmatrix}
\]

A diagonal matrix is its own transpose: $D^T = D$. In general, we find

\[a_{ij} \iff a_{ji}\]

Symmetric matrix is a square matrix which is symmetrical about its principal diagonal, so $a_{ij} = a_{ji}$. Therefore a symmetrical matrix is equal to its own transpose.

\[
A = A^T = \begin{bmatrix}
1 & 2 & 5 \\
2 & 6 & 3 \\
5 & 3 & 4
\end{bmatrix} = \text{symmetric}
\]

Skew symmetric matrix is a matrix in which:

\[a_{ij} = -a_{ji}\]

So

\[
A^T = -A
\]

\[a_{ii} = 0\text{(principal diagonal elements are zero)}\]

\[
A = \begin{bmatrix}
0 & 4 & -5 \\
-4 & 0 & 3 \\
5 & -3 & 0
\end{bmatrix} = \text{skew symmetric}
\]

Any square matrix can be split into the sum of a symmetric and skew symmetric matrix:

\[
A = \frac{1}{2}(A + A^T) + \frac{1}{2}(A - A^T)
\]

Basic Matrix Operations:

Matrix addition and subtraction requires matrices of the same order since this operation simply involves addition or subtraction of corresponding elements. So, if $A + B = C$,

\[
A = \begin{bmatrix}
a_{11} & a_{12} \\
a_{21} & a_{22} \\
a_{31} & a_{32}
\end{bmatrix}; B = \begin{bmatrix}
b_{11} & b_{12} \\
b_{21} & b_{22} \\
b_{31} & b_{32}
\end{bmatrix}; C = \begin{bmatrix}
a_{11} + b_{11} & a_{12} + b_{12} \\
a_{21} + b_{21} & a_{22} + b_{22} \\
a_{31} + b_{31} & a_{32} + b_{32}
\end{bmatrix}
\]

and

\[
(1) A + B = B + A
\]

\[
(2)(A + B) + C = A + (B + C)
\]

where all matrices are the same order. Scalar multiplication of a matrix is multiplying a matrix by a constant (scalar):
where $\beta$ is a scalar. Every element is multiplied by the scalar. Scalar product (or dot product or inner product) is the product of 2 vectors of the same size.

$$a \cdot b = \beta$$

where $a$ is a row vector (or the transpose of a column vector) of length $n$, $b$ is a column vector (or the transpose of a row vector), also of length $n$, and $\beta$ is the scalar product of $a \cdot b$. Then:

$$a = [a_1 \ a_2 \ a_3]; \quad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

and

$$\beta = a_1b_1 + a_2b_2 + a_3b_3$$

Some people like to visualize this multiplication as:

Conceptually, this product can be thought of as multiplying the length of one vector by the component of the other vector which is parallel to the first:

$$a \cdot b = |a||b| \cos(\theta)$$

where
\[ |x| = \sqrt{x_1^2 + x_2^2 + \ldots + x^2} \]

The maximum principle says that the unit vector \((n)\) making \(a \cdot n\) a maximum is that unit vector pointing in the same direction as \(a\): If \(n \parallel a\) then \(\cos(\theta) = \cos(0^\circ) = 1\) and \(a \cdot n = |a| |n| \cos(\theta) = |a||n| = |a|\). This is equally true where \(d\) is any vector of a given magnitude - that vector \(n\) which parallels \(d\) will give the largest scalar product.

Parallel vectors thus have \(\cos(\theta) = 1\), then \(a \cdot b = |a| |b|\) and \(a = \beta b\) (i.e., 2 vectors are parallel if one is simply a scalar multiple of the other - this property comes from equating direction cosines) where:

\[ \beta = \frac{|a|}{|b|} \]

Perpendicular vectors have \(\cos(\theta) = \cos 90^\circ = 0\), so \(a \cdot b = 0\), where \(a \perp b\). Squaring vectors is simply:

\[ a^2 = a \cdot a^T \text{ for rowvectors} \]

\[ a^2 = a^T \cdot a \text{ for columnvectors} \]

Matrix multiplication requires "conformable" matrices. Conformable matrices are ones in which there are as many columns in the first as there are rows in the second:

\[ C_{(m,n)} = A_{(m,p)} \cdot B_{(p,n)} \]

So, the product matrix \(C\) is of order \(m \times n\) and has elements \(c_{ij}\):

\[ c_{ij} = \sum_{k=1}^{p} a_{ik}b_{kj} \]

This is extension of scalar product - in this case, each element of \(C\) is the scalar product of a row vector in \(A\) and column vector in \(B\).

\[
\begin{bmatrix}
  c_{11} & c_{12} \\
  c_{21} & c_{22}
\end{bmatrix}
= \begin{bmatrix}
  a_{11} & a_{12} & a_{13} \\
  a_{21} & a_{22} & a_{23}
\end{bmatrix}
\begin{bmatrix}
  b_{11} & b_{12} \\
  b_{21} & b_{22} \\
  b_{31} & b_{32}
\end{bmatrix}
\]

\[ c_{12} = a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} \]

In "box form":

\[
\begin{bmatrix}
  c_{11} & c_{12} \\
  c_{21} & c_{22}
\end{bmatrix}
= \begin{bmatrix}
  a_{11} & a_{12} & a_{13} \\
  a_{21} & a_{22} & a_{23}
\end{bmatrix}
\begin{bmatrix}
  b_{11} & b_{12} \\
  b_{21} & b_{22} \\
  b_{31} & b_{32}
\end{bmatrix}
\]
Elements of \( C \) are scalar products of corresponding vectors as shown here.

Fig. 3-3. The matrix product of two matrices.

Order of multiplication is important—usually

\[ A \cdot B \neq B \cdot A \]

and unless \( A \) and \( B \) are square matrices or the order of \( A^T \) is the same as the order of \( B \) (or vice versa), one of the two products can not even be formed. Order is specified by stating:

\( A \) is \textbf{pre}-multiplied by \( B \) (for \( B \cdot A \))

\( A \) is \textbf{post}-multiplied by \( B \) (for \( A \cdot B \))

Multiple products:

\[ D = \left( A \cdot B \right) \cdot C = A \cdot \left( B \cdot C \right) \]

(The order in which the pairs are multiplied is not important mathematically.)

Computational considerations:

\[ C_{(mn)} = A_{(mp)} \cdot B_{(pn)} \]

involves \( m \times n \times p \) multiplications and \( m \times n \times (p-1) \) additions, so:

\[ E_{(m,n)} = [A_{(mp)} \cdot B_{(pq)}] \cdot C_{(q,n)} \]

gives \( m \times p \times q \) multiplications = \( D_{(mq)} \cdot C_{(qn)} \)

gives \( m \times q \times n \) multiplications

and

\[ E_{(m,n)} = A_{(mp)} \cdot [B_{(pq)} \cdot C_{(qn)}] \]

gives \( p \times q \times n \) multiplications = \( A_{(mp)} \cdot D_{(pq)} \)

gives \( m \times p \times n \) multiplications
Therefore:

1) \((A \cdot B) \cdot C \Rightarrow mq(p + n)\) total multiplications
2) \(A \cdot (B \cdot C) \Rightarrow pn(m + q)\) total multiplications

If both \(A\) and \(B\) are 100 x 100 matrices and \(C\) is 100 x 1, then \(m = 100, p = 100, q = 100\), and \(n = 1\). Multiplying using form 1) involves \(\approx 1 \times 10^6\) multiplications, whereas form 2) involves \(2 \times 10^4\); so computing \(B \cdot C\) first, then pre-multiplying by \(A\) saves almost a million multiplications and almost an equal number of additions in this example. Therefore order is extremely important computationally for both speed and accuracy (more operations lead to a greater accumulation of round-off errors).

Transpose of matrix product is simply multiplication by transpose of the individual matrices in reverse order:

\[
D = A \cdot B \cdot C
\]

\[
D^T = C^T \cdot B^T \cdot A^T
\]

Multiplication by \(I\) leaves the matrix unchanged:

\[
A \cdot I = I \cdot A = A
\]

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

Pre-multiplication by a diagonal matrix:

\[
C = D \cdot A\]

where \(D\) is a diagonal matrix, \(C\) is the \(A\) matrix with each row scaled by a diagonal element of \(D\):

\[
D = \begin{bmatrix}
d_{11} \\
d_{22} \\
d_{33}
\end{bmatrix}; \quad A = \begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
 a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{bmatrix}
\]

\[
C = \begin{bmatrix}
a_{11}d_{11} & a_{12}d_{11} & a_{13}d_{11} \\
a_{21}d_{22} & a_{22}d_{22} & a_{23}d_{22} \\
a_{31}d_{33} & a_{32}d_{33} & a_{33}d_{33}
\end{bmatrix} \quad \leftarrow \text{each element} \times d_{11}
\]

\[
\begin{bmatrix}
a_{11}d_{11} & a_{12}d_{11} & a_{13}d_{11} \\
a_{21}d_{22} & a_{22}d_{22} & a_{23}d_{22} \\
a_{31}d_{33} & a_{32}d_{33} & a_{33}d_{33}
\end{bmatrix} \quad \leftarrow \text{each element} \times d_{22}
\]

\[
\begin{bmatrix}
a_{11}d_{11} & a_{12}d_{11} & a_{13}d_{11} \\
a_{21}d_{22} & a_{22}d_{22} & a_{23}d_{22} \\
a_{31}d_{33} & a_{32}d_{33} & a_{33}d_{33}
\end{bmatrix} \quad \leftarrow \text{each element} \times d_{33}
\]

Post-multiplication by a diagonal matrix produces a matrix in which each column has been scaled by a diagonal element.

\[
C = A \cdot D
\]

\[
C = \begin{bmatrix}
a_{11}d_{11} & a_{12}d_{22} & a_{13}d_{33} \\
a_{21}d_{11} & a_{22}d_{22} & a_{23}d_{33} \\
a_{31}d_{11} & a_{32}d_{22} & a_{33}d_{33}
\end{bmatrix}
\]

where each column in \(A\) has been scaled by the corresponding diagonal matrix elements, \(d_{ii}\).

The determinant of a matrix is a single number representing a property of a square matrix.
GG313 GEOLOGICAL DATA ANALYSIS 3–8

(dependent upon what the matrix represents). The main use here is for finding the inverse of a matrix or solving simultaneous equations. Symbolically, the determinant is usually given as det $A$, $|A|$ or $\|A\|$ (to differentiate from magnitude). Calculation of a 2 x 2 determinant is given by:

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

This is the difference of the cross products. The calculation of an $n \times n$ determinant is given by

$$\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix} = a_{11}m_{11} - a_{12}m_{12} + a_{13}m_{13} - \cdots - (-1)^n a_{1n}m_{1n},$$

where $m_{11}$ is the determinant with the first row and column missing; $m_{12}$ is the determinant with the first row and second column missing; etc. (The determinant of a 1 x 1 matrix is just the particular element.) An example of a 3 x 3 determinant follows.

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

$$m_{11} = \begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} = a_{22}a_{33} - a_{23}a_{32}$$

$$m_{12} = \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} = a_{21}a_{33} - a_{23}a_{31}$$

$$m_{13} = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} = a_{21}a_{32} - a_{22}a_{31}$$

So

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}m_{11} - a_{12}m_{12} + a_{13}m_{13}$$

For a 4 x 4 determinant, each $m_{ij}$ would be an entire expansion given above for the 3 x 3 determinant—one quickly needs a computer.

A singular matrix is a square matrix whose determinant is zero. A determinant is zero if:

1) any row or column is zero;
2) any row or column is equal to a linear combination of the other rows or columns. For example:

$$\begin{vmatrix} 1 & 6 & 4 \\ 2 & 1 & 0 \\ 5 & -3 & -4 \end{vmatrix}$$

where row 1 = 3(row 2) - row 3.
\[ A = \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix} \]

The degree of clustering symmetrically about the principal diagonal is another (of many) properties of a determinant. The more the clustering, the higher the value of the determinant.

Matrix division can be thought of as multiplying by the inverse. Consider scalar division:
\[
\frac{x}{b} = x_b^{-1} = xb^{-1}
\]
which we can write
\[
bb^{-1} = 1
\]
Matrices can be effectively divided by multiplying by the inverse matrix. Nonsingular square matrices may have an inverse symbolized as \( A^{-1} \) and \( AA^{-1} = 1 \). The calculation of matrix inverse is usually done using elimination methods on the computer. For a simple 2 x 2 matrix, its inverse is given by:

\[
A^{-1} = \frac{1}{|A|} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}
\]

An example follows.
\[
A = \begin{bmatrix} 7 & 2 \\ 10 & 3 \end{bmatrix};
\]
\[
A^{-1} = \frac{1}{21 - 20} \begin{bmatrix} 3 & -2 \\ -10 & 7 \end{bmatrix} = \begin{bmatrix} 3 & -2 \\ -10 & 7 \end{bmatrix}
\]
\[
AA^{-1} = \begin{bmatrix} 7 & 2 \\ 10 & 3 \end{bmatrix} \begin{bmatrix} 3 & -2 \\ -10 & 7 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I
\]

Solution of Simultaneous Equations

A system of \( n \) simultaneous equations in \( n \) unknowns:
\[
\begin{align*}
        a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= b_1 \\
        a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 &= b_2 \\
        a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 &= b_3 \\
        a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 &= b_4
\end{align*}
\]
can be written as
\[
Ax = b
\]
where
\[
A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \text{coefficient matrix}
\]
Then
\[ A^{-1}Ax = A^{-1}b \] (pre-multiplying both sides by \( A^{-1} \))
so
\[ Ix = x = A^{-1}b \]
gives the solution for values of \( x_1, x_2, x_3, x_4 \) which solve the system. The following example solves for 2 simultaneous equations. Consider 2 equations in 2 unknowns (e.g., equations of lines in the \( x-y \) plane):

\[
\begin{align*}
5x_1 + 7x_2 &= 19 \\
3x_1 - 2x_2 &= -1
\end{align*}
\]

In matrix form this translates to:
\[
\begin{bmatrix} 5 & 7 \\ 3 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 19 \\ -1 \end{bmatrix}
\]

To solve this matrix, we need the inverse of \( A \):

\[
A^{-1} = \frac{1}{-10 -21} \begin{bmatrix} -2 & -7 \\ -3 & 5 \end{bmatrix} = \begin{bmatrix} 2 & \frac{7}{31} \\ \frac{3}{31} & 3 \end{bmatrix}
\]

Then \( x = A^{-1}b \) where

\[
A^{-1}b = \begin{bmatrix} 2 & \frac{7}{31} \\ \frac{3}{31} & -5 \end{bmatrix} \begin{bmatrix} 19 \\ -1 \end{bmatrix} = \begin{bmatrix} \frac{38}{31} - \frac{7}{31} \\ \frac{57}{31} + \frac{5}{31} \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}
\]
or, in box form:

\[
A^{-1} = \begin{bmatrix} 2 & \frac{7}{31} \\ \frac{3}{31} & -5 \end{bmatrix} \quad \begin{bmatrix} 19 \\ -1 \end{bmatrix} \cdot b = \begin{bmatrix} 1 \\ 2 \end{bmatrix} 
\]

So, the values \( x_1 = 1 \) and \( x_2 = 2 \) solve the above system, or
Computational considerations

While this approach may seem burdensome, it is good because it is extremely general and allows a simple handling and a straightforward solution to very large systems. However, it is true that direct (elimination) methods to the solution are in fact quicker for fully populated matrices:

1) Solution to inverse matrix approach involves $n^3$ multiplications for the inversion and $n^2m$ more multiplications to finish the solution, where $n$ is the number of equations per set, and $m$ is the number of sets of equations (each of the same form but different $b$ matrix). The total number of multiplications is $n^3 + n^2m$.

2) Solution to directly solving equations involves $n^3/3 + n^2m$.

So, while the matrix form is easy to handle, one should not necessarily always use it blindly. We will consider many situations for which matrix solutions are ideal. For sparse or symmetrical matrices, the above relationships may not hold. The rank of a matrix is the number of linearly independent vectors it contains (either row or column vectors): 

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

Since row 3 = -(row 1) - 2(row 2) or col 3 = col 1 - 1/4(col 2) and col 4 = -(col 1) + 3/4(col 2), the matrix $A$ has rank 2 (i.e., it has only 2 linearly independent vectors, independent of whether viewed by rows or columns). The rank of a matrix product must be less than or equal to the smallest rank of the matrices being multiplied. 

\[
A_{(\text{rank} 2)} \cdot B_{(\text{rank} 1)} = C_{(\text{rank} 1)}
\]

Therefore, (from another angle), if a matrix has rank $r$, than any matrix factor of it must have rank of at least $r$. Since the rank cannot be greater than the smallest of $m$ or $n$, in a $m \times n$ matrix, this definition also limits the size (order) of factor matrices. (That is, one cannot factor a matrix of rank 2, into 2 matrices of which either is of less than rank 2, so $m$ and $n$ of each factor must also be $\geq 2$).

The trace of a square matrix is simply the sum of the elements along the principal diagonal. It is symbolized as $\text{tr } A$. This property is useful in calculating various quantities from matrices. Submatrices are smaller matrix partitions of the larger supermatrix:

\[
\begin{bmatrix}
\text{Supermatrix} \\
F
\end{bmatrix} = \begin{bmatrix}
A & B \\
C & D
\end{bmatrix}
\]

Such partitioning will frequently be useful. Other useful matrix properties:
1. \((A^T)^T = A\)
2. \((A^{-1})^{-1} = A\)
3. \((A^{-1})^T = (A^T)^{-1} = A^{-T}\)
4. \(D = ABC\), then \(D^{-1} = C^{-1} B^{-1} A^{-1}\); recall that \(D = ABC\); \(D^T = C^T B^T A^T\)

This "reversal rule" for inverse products may be useful for eliminating or minimizing the number of matrix inverses requiring calculation.

We will look at a few examples of matrix manipulations. For data matrix \(A\):

\[
A = \begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{bmatrix}
\]

and unit row vector \(j\):

\[
j = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}
\]

(1) Compute the mean of each column vector in \(A\) (each column has length \(n = 3\)):

\[
\bar{x}_c = \frac{1}{3} j A = j_n A
\]

Then

\[
j_n : \begin{bmatrix} 1 \\
4 \\
7
\end{bmatrix} = \begin{bmatrix} 1 \\
4 \\
7
\end{bmatrix}
\]

\[
j_n : \begin{bmatrix} 1 \\
3 \\
3
\end{bmatrix} = \begin{bmatrix} 1 \\
3 \\
3
\end{bmatrix}
\]

\[
j_n : \begin{bmatrix} 1 \\
3 \\
3
\end{bmatrix} = \begin{bmatrix} 1 \\
3 \\
3
\end{bmatrix}
\]

(2) Compute the mean of each row vector in \(A\):

\[
\bar{x}_r = \frac{1}{3} A j^T = A j_n^T
\]

Then

\[
\begin{bmatrix} 1/3 \\
1/3 \\
1/3
\end{bmatrix} : j_n^T
\]

\[
A : \begin{bmatrix} 1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{bmatrix} = \begin{bmatrix} 2 \\
5 \\
8
\end{bmatrix} : \bar{x}_r
\]

Last time we looked at the matrix equation

\[A \ x = b\]

and we found that the solution could be written
\[ x = A^{-1} b \]

For a moment, let us just consider the left hand side \( A \cdot x \). For any \( x \), this product gives a new vector \( y \). We can say that \( x \) is transformed to give \( y \). This is a linear transformation since there are only linear terms in the matrix multiplication, i.e., the vector \( y \) is

\[
\begin{bmatrix}
    a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \\
    a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \\
    a_{31}x_1 + a_{32}x_2 + a_{33}x_3
\end{bmatrix}
\]

Thus we call the operation \( T(x) = A \cdot x \) a linear transformation. If we stick to three or less dimensions, it is possible to graphically visualize vectors and operations on them. Figure 3-4 shows an arbitrary vector \( x \) and the result \( y \) of the linear transformation \( y = A \cdot x \).

![Graphical representation of linear transformation](image)

Fig. 3-4. The vector \( x \) is transformed into another vector \( y \) using a linear transformation.

Obviously, as we pick another \( x \), we get another \( y \). We might want to know if there are certain vectors that, when being operated on, returns a vector in the same direction, possibly longer or shorter than the original \( x \). In other words, are there an \( x \) that satisfies

\[ Ax = \lambda x \]  \hspace{1cm} (3.1)

We call \( \lambda \) the eigenvalue and \( x \) the eigenvector. We can rewrite this as

\[
\begin{align*}
    A x - \lambda I x &= 0 \\
    (A - \lambda I)x &= 0
\end{align*}
\]

or

\[
B x = 0 = n
\]

In general, the solution to this equation can be written \( x = B^{-1} n \):

\[
x = \frac{1}{|B|} \begin{bmatrix} \cdots & \cdots & \cdots \end{bmatrix} n = B^{-1} n
\]

or
So apart from the trivial solution $x = [0 \ 0 \ 0]$, the answer is given when

$$|B| = 0$$

We know the determinant of $B$ is

$$|B| = \begin{vmatrix} a_{11} - \lambda & a_{12} & a_{13} \\ a_{21} & a_{22} - \lambda & a_{23} \\ a_{31} & a_{32} & a_{33} - \lambda \end{vmatrix}$$

Writing out what the determinant is and setting it to zero gives a polynomial in $\lambda$ of order $n$. For $n = 3$ this will in general give a cubic equation; for $n = 2$ a quadratic equation must be solved. The solutions $\lambda_1$, $\lambda_2$, ..., etc. are called the eigenvalues of $A$, and the equation $|B| = 0$ is called the characteristic equation. For example, given

$$A = \begin{bmatrix} 17 & -6 \\ 45 & -16 \end{bmatrix}$$

let us find its eigenvalues. We set

$$|A - \lambda I| = \begin{vmatrix} 17 - \lambda & -6 \\ 45 & -16 - \lambda \end{vmatrix} = 0$$

or

$$-272 - 17\lambda + 16\lambda + \lambda^2 + 270 = \lambda^2 - \lambda - 2 = 0$$

We easily solve for $\lambda$:

$$\lambda = \frac{1 \pm \sqrt{1^2 - 4(-2)}}{2} = 2, -1$$

So the eigenvalues are $\lambda_1 = 2$, $\lambda_2 = -1$. We now know what $\lambda$ must be for (3.1) to be satisfied, but what about the vectors $x$? We still haven't found what they must be, but we will substitute in the value for $\lambda$ in (3.1). Using $\lambda = 2$ first, we find

$$A x = 2 x$$

$$(A - 2 I ) x = 0$$

Find

$$\begin{bmatrix} 15 & -6 \\ 45 & -18 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

or

$$15x_1 - 6x_2 = 0$$

$$45x_1 - 18x_2 = 0$$

which both give
\[ x_1 = \frac{2}{5} x_2 \]

So
\[ \mathbf{x} = \begin{bmatrix} 2 \\ 5 \end{bmatrix} \]

where \( t \) is any scalar. Similarly, for \( \lambda = -1 \), we find \((A + I)x = 0\)
\[
\begin{bmatrix} 18 & -6 \\ 45 & -15 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]

which reduces to
\[ 3x_1 - x_2 = 0 \]

which gives
\[ \mathbf{x} = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \]

It may happen that the characteristic equation gives solutions that are imaginary. However, if the matrix is symmetric it will always yield real eigenvalues, and as long as the matrix \( A \) is not singular, all the \( \lambda \) will be non-zero and the corresponding eigenvectors will be orthogonal.

The technique we've used applies to matrices of any size \( n \times n \), but finding the roots of large polynomials is painful. Usually, the \( \lambda \) are found by matrix manipulations that involve successive approximations to the \( x \). This is of course only practical on a computer. If we restricted our attention to 2-D geometry, certain properties of eigenvalues and eigenvectors will be clearer.

Consider the matrix \( A \)
\[
A = \begin{bmatrix} 4 & 8 \\ 8 & 4 \end{bmatrix}
\]

Fig. 3-5. Graphical representation of two vectors in the x-y plane.

We can regard the matrix as two row vectors \([4 \ 8]\) and \([8 \ 4]\). Let us find the eigenvalues and eigenvectors of \( A \):
\[
\begin{vmatrix}
4 - \lambda & 8 \\
8 & 4 - \lambda
\end{vmatrix} = 0
\]

\[16 - 8\lambda + \lambda^2 - 64 = 0\]
\[\lambda^2 - 8\lambda - 48 = 0\]
\[\lambda = \frac{8 \pm \sqrt{64 + 4 \cdot 48}}{2} = 12, -4\]

The eigenvectors are:

\[
\begin{bmatrix}
4 - 12 & 8 \\
8 & 4 - 12
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix} = \begin{bmatrix}
-8 & 8 \\
8 & -8
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix} = \begin{bmatrix}
0 \\
0
\end{bmatrix}
\]

\[-8x_1 + 8x_2 = 0 \Rightarrow x_1 = x_2 \quad e_1 = \begin{bmatrix} 1 & 1 \end{bmatrix}\]

\[
\begin{bmatrix}
4 + 4 & 8 \\
8 & 4 + 4
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix} = \begin{bmatrix}
8 & 8 \\
8 & 8
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix} = \begin{bmatrix}
0 \\
0
\end{bmatrix}
\]

\[8x_1 + 8x_2 = 0 \Rightarrow x_1 = -x_2 \quad e_2 = \begin{bmatrix} -1 & 1 \end{bmatrix}\]

We find that the eigenvectors define the minor and major axis of the ellipse which goes through the two points defined by (4,8) and (8,4). The length of these axes are given by the absolute values of the eigenvalues, \(\lambda_1 = 12\) and \(\lambda_2 = 4\).

Fig. 3-6. The eigenvectors, scaled by the eigenvalues, can be seen to represent the major and minor axes of the ellipse that goes through the two data vectors (8, 4) and (4, 8).
It is customary to normalize the eigenvectors so that their length is unity. In our case we find

\[ e_1^T = \begin{bmatrix} \frac{\sigma_1}{2} \\ \frac{\sigma_2}{2} \end{bmatrix} \quad \text{and} \quad e_2^T = \begin{bmatrix} -\frac{\sigma_1}{2} \\ \frac{\sigma_2}{2} \end{bmatrix} \]

The axes of the ellipse are then simply

\[ v_1 = \lambda_1 e_1 \]
\[ v_2 = \lambda_2 e_2 \]

Since the sign of the eigenvector is indeterminate we choose to make all eigenvalues positive and thus place the minus-sign in \( \lambda_2 \) inside \( e_2 \). You'll notice that \( v_1 \cdot v_2 = 0 \), i.e., they are orthogonal. The eigenvectors make up the columns in a new matrix \( V \)

\[ V = \begin{bmatrix} e_1 & e_2 \end{bmatrix} = \frac{\sigma_2}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \]

Let us expand the eigenvalue equation (3.1) \( A \cdot x = \lambda x \) to a full matrix equation. We have

\[ A \cdot e_1 = \lambda_1 e_1 \]
\[ A \cdot e_2 = \lambda_2 e_2 \]

We can combine these two matrix equations into one.

\[ A \cdot V = V \cdot \Lambda \]

where

\[ \Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \]

From this we may learn two things. 1) Post multiply by \( V^{-1} \):

\[ A \cdot V \cdot V^{-1} = A = V \cdot \Lambda \cdot V^{-1} \]

The eigenvalue-eigenvector operation let us split a symmetric matrix \( A \) into a diagonal matrix (with the eigenvalues along the diagonal) and the matrix \( V \) (with the eigenvectors as rows) and its inverse \( V^{-1} \).

2) Pre-multiplying by \( V^{-1} \):

\[ \Lambda = V^{-1} \cdot A \cdot V \]

This operation transforms the \( A \) matrix into a diagonal matrix \( \Lambda \). It corresponds to a rotation of the coordinate axes in which the eigenvectors in \( V \) becomes the new coordinate axes. Relative to the new coordinates, \( \Lambda \) conveys the same information as \( A \) does in the old coordinates. Because \( \Lambda \) is a simple diagonal matrix, the rotation (transformation) makes the relationships between rows and columns in \( A \) much clearer.

**Simple Regression and Curve Fitting**

Whereas an interpolant fits each data point exactly, it is frequently advantageous to produce a smoothed fit to the data - not exactly fitting each point, but producing a "best" fit. A popular (and
convenient) method for producing such fits is known as the **method of least squares**.

The method of least squares produces a fit of a specified (usually continuous) basis to a set of data points which minimizes the sum of the squared mismatch (error) between the fitted curve and data. The error can be measured as in Figure 13-1:

This regression of $y$ on $x$ is the most common method. Less common methods (more work involved) is regression of $x$ on $y$ and orthogonal regression (which we will return to later).

![Fig. 3-7. Graphical representation of the regression errors used in least-squares procedures.](image)

![Fig. 3-8. Two other regression methods: regressing $x$ on $y$ and orthogonal regression.](image)

**Least squares simple example**

Consider fitting a single "best" linear slope to $n$ data points. This can be a scatter plot of $y(t)$, $x(t)$ plotted at similar values of $t$; or a simple $f(x)$ relationship. At any rate, $y$ is considered a function of $x$. We wish to fit a line of the form

$$y = a_1 + a_2(x - x_0)$$  \hspace{1cm} (3.2)

and must therefore determine a value for $a_1$ and $a_2$ which produces a line that minimizes the sum of the squared errors ($x_0$ is specified beforehand). So

$$\text{minimize } \sum_{i=1}^{n} (y_{\text{computed}} - y_{\text{observed}})^2$$

Ideally, for each observation $y_i$ at $x_i$ we should have
There are many more equations \((n - 1)\) than unknowns \((2 - a_1 \text{ and } a_2)\). Such a system is over-determined and there exists no unique solution (unless all the \(y_i\)’s do lie exactly on a single line, in which case any two equations will uniquely determine \(a_1\) and \(a_2\)).

In matrix form (i.e., \(Ax = b\)):

\[
\begin{bmatrix}
1 & (x_1 - x_0) \\
1 & (x_2 - x_0) \\
\vdots & \vdots \\
1 & (x_n - x_0)
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_2
\end{bmatrix}
= \begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
y_n
\end{bmatrix}
\]

Since \(A\) is a non-square matrix it has no inverse, so the equation cannot be inverted and solved. Consider instead the error in the fit at each point:

\[
\begin{align*}
a_1 + a_2(x_i - x_0) - y_i &= e_i \\
a_1 + a_2(x_2 - x_0) - y_2 &= e_2 \\
\vdots \\
a_1 + a_2(x_n - x_0) - y_n &= e_n
\end{align*}
\]

We wish to determine the values for \(a_1\) and \(a_2\) that minimize

\[
\sum_{i=1}^{n} e_i^2
\]

This will minimize the variance of the residuals about the regression line and give the least-squares fit. Notation:

\[
S = E(a_1, a_2) = \sum_{i=1}^{n} e_i^2 \quad (= e^T e \quad \text{in matrix form})
\]

So, \(E(a_1, a_2)\) and the minimum of this function (with respect to the two unknown coefficients) can be determined using simple differential calculus, where

\[
\frac{\partial E(a_1, a_2)}{\partial a_1} = \frac{\partial E(a_1, a_2)}{\partial a_2} = 0 \quad \text{(at the minimum)}
\]

\[
\frac{\partial E}{\partial a_1} = \frac{\partial}{\partial a_1} \left( \sum_{i=1}^{n} e_i^2 \right) = \frac{\partial}{\partial a_1} \left( \sum_{i=1}^{n} \left[ a_1 + a_2(x_i - x_0) - y_i \right]^2 \right)
\]

\[
= 2 \left\{ \sum_{i=1}^{n} \left[ a_1 + a_2(x_i - x_0) - y_i \right] \right\} = 0
\]

\[
\frac{\partial E}{\partial a_2} = \frac{\partial}{\partial a_2} \left( \sum_{i=1}^{n} e_i^2 \right) = \frac{\partial}{\partial a_2} \left( \sum_{i=1}^{n} \left[ a_1 + a_2(x_i - x_0) - y_i \right]^2 \right)
\]
These two equations can now be expanded into their individual terms forming what are known as the **normal equations**:

\[
\sum_{i=1}^{n} a_1 + \sum_{i=1}^{n} a_2 (x_i - x_0) = \sum_{i=1}^{n} y_i \\
\sum_{i=1}^{n} a_1 (x_i - x_0) + \sum_{i=1}^{n} a_2 (x_i - x_0)^2 = \sum_{i=1}^{n} y(x_i - x_0)
\]

The normal equations thus provide a system of 2 equations in 2 unknowns which can be uniquely solved. Rearranging,

\[
n a_1 + a_2 \sum_{i=1}^{n} (x_i - x_0) = \sum_{i=1}^{n} y_i \\
a_1 \sum_{i=1}^{n} (x_i - x_0) + a_2 \sum_{i=1}^{n} (x_i - x_0)^2 = \sum_{i=1}^{n} y(x_i - x_0)
\]

Notice that all sums are sums of known values that sum to simple constants. Solving:

\[
a_1 = \frac{1}{n} \sum_{i=1}^{n} y_i - \frac{a_2}{n} \sum_{i=1}^{n} (x_i - x_0)
\]

substitute this into the second equation:

\[
\left[ \frac{1}{n} \sum_{i=1}^{n} y_i - \frac{a_2}{n} \sum_{i=1}^{n} (x_i - x_0) \right] \sum_{i=1}^{n} (x_i - x_0) + a_2 \sum_{i=1}^{n} (x_i - x_0)^2 = \sum_{i=1}^{n} y(x_i - x_0)
\]

Now solve for \(a_2\):

\[
\frac{1}{n} \sum_{i=1}^{n} y_i \sum_{i=1}^{n} (x_i - x_0) - \frac{a_2}{n} \left[ \sum_{i=1}^{n} (x_i - x_0) \right]^2 + a_2 \sum_{i=1}^{n} (x_i - x_0)^2 = \sum_{i=1}^{n} y(x_i - x_0)
\]

\[
a_2 \left\{ \sum_{i=1}^{n} (x_i - x_0)^2 - \frac{1}{n} \left[ \sum_{i=1}^{n} (x_i - x_0) \right]^2 \right\} = \sum_{i=1}^{n} y(x_i - x_0) - \frac{1}{n} \sum_{i=1}^{n} y_i \sum_{i=1}^{n} (x_i - x_0)
\]

Finally,

\[
a_2 = \left[ \sum_{i=1}^{n} y_i (x_i - x_0) - \frac{1}{n} \sum_{i=1}^{n} y_i \sum_{i=1}^{n} (x_i - x_0) \right] / \left\{ \sum_{i=1}^{n} (x_i - x_0)^2 - \frac{1}{n} \left[ \sum_{i=1}^{n} (x_i - x_0) \right]^2 \right\}
\]

Substitute \(a_2\) into the first equation to solve for \(a_1\). So, we compute the sums

\[
\sum_{i=1}^{n} y_i, \sum_{i=1}^{n} y(x_i - x_0), \sum_{i=1}^{n} (x_i - x_0), \text{ and } \sum_{i=1}^{n} (x_i - x_0)^2
\]

and substitute into the above equation to give \(a_1\) and \(a_2\) producing the best fit. In matrix form the normal equations are:

\[
\begin{bmatrix}
\sum_{i=1}^{n} (x_i - x_0) & \sum_{i=1}^{n} (x_i - x_0)^2 \\
\sum_{i=1}^{n} (x_i - x_0) & \sum_{i=1}^{n} (x_i - x_0)^2
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_2
\end{bmatrix} =
\begin{bmatrix}
\sum_{i=1}^{n} y_i \\
\sum_{i=1}^{n} y(x_i - x_0)
\end{bmatrix}
\]

(3.5)
So, $Nx = B$, e.g., of the form $Ax = b$. Since $N$ is square and of full rank, this equation is solved in the standard manner:

$$N^{-1}Nx = N^{-1}B$$

or

$$x = N^{-1}B$$

This problem was simple enough (2 x 2) to solve brute force for $a_1$ and $a_2$. For larger systems this becomes impractical and a matrix solution to the non-square $A \cdot x = b$ equation must be sought. We will next look at the general linear least-squares problem and find a solution in matrix notation.

**General Linear Least Squares**

We have looked at a few special cases where we have sought to fit a "model" to "data" in a least-squares sense. Fitting a straight line to the $x$-$y$ points was a very simple example of this technique. We will now look at the more general problem of finding the coefficients for any linear combination of basis functions that fits some data in a least squares sense. There are numerous situations where this is needed:

<table>
<thead>
<tr>
<th>Situation</th>
<th>Model</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Curve Fitting</td>
<td>Coefficients of polynomials, sin/cos, etc.</td>
<td>Points in x-y plane</td>
</tr>
<tr>
<td>Gravity modeling</td>
<td>Densities of subsurface polygons</td>
<td>Gravity observations</td>
</tr>
<tr>
<td>Hypocenter location</td>
<td>Small pertubations to hypocenter position</td>
<td>Seismic arrival times</td>
</tr>
</tbody>
</table>

While the basis functions in these cases are all vastly different, they are all used in a linear combination to fit the observed data. We will therefore take time to investigate how such a problem is set up, and how it can all be simplified with matrix algebra.

**General (linear) least squares.**

Consider the least squares fitting of any continuous basis of the form

$$x_1, x_2, x_3, \ldots, x_m$$

For example

**polynomial basis**

- $x_1 = x^0$
- $x_2 = x^1$
- $x_3 = x^2$

**Fourier sine basis**

- $x_1 = \sin \left( \frac{2\pi x}{T} \right)$
- $x_2 = \sin \left( \frac{4\pi x}{T} \right)$
- $x_3 = \sin \left( \frac{6\pi x}{T} \right)$
We desire to fit an equation of the form

\[ y = a_1 x_1 + a_2 x_2 + \cdots + a_m x_m \]

to a data set of \( n \) data points, where \( n > m \), by minimizing \( E \):

\[
E = \sum_{i=1}^{n} (e_i)^2 = \sum_{i=1}^{n} (a_1 x_{i1} + a_2 x_{i2} + \cdots + a_m x_{im} - y_i)^2
\]

(3.6)

where \( y_i \) is the observed value. We can write this explicitly:

\[
\begin{align*}
  a_1 x_{11} + a_2 x_{12} + \cdots + a_m x_{1m} - y_1 &= e_1 \\
  a_1 x_{21} + a_2 x_{22} + \cdots + a_m x_{2m} - y_2 &= e_2 \\
  &\vdots \\
  a_1 x_{n1} + a_2 x_{n2} + \cdots + a_m x_{nm} - y_n &= e_n
\end{align*}
\]

where \( x_{ij} \) is the \( j \)'th \( x \) function of the basis, evaluated at the value \( x_i \). To minimize \( E \), we set

\[
\frac{\partial E(a_j)}{\partial a_j} = 0
\]

(3.7)

So

\[
\begin{align*}
\frac{\partial E(a_1)}{\partial a_1} &= \frac{\partial}{\partial a_1} \sum_{i=1}^{n} (a_1 x_{i1} + a_2 x_{i2} + \cdots + a_m x_{im} - y_i)^2 \\
&= 2 \sum_{i=1}^{n} (a_1 x_{i1} + a_2 x_{i2} + \cdots + a_m x_{im} - y_i) x_{i1} = 0 \\

\frac{\partial E(a_2)}{\partial a_2} &= \frac{\partial}{\partial a_2} \sum_{i=1}^{n} (a_1 x_{i1} + a_2 x_{i2} + \cdots + a_m x_{im} - y_i)^2 \\
&= 2 \sum_{i=1}^{n} (a_1 x_{i1} + a_2 x_{i2} + \cdots + a_m x_{im} - y_i) x_{i2} = 0 \\

\frac{\partial E(a_m)}{\partial a_m} &= \frac{\partial}{\partial a_m} \sum_{i=1}^{n} (a_1 x_{i1} + a_2 x_{i2} + \cdots + a_m x_{im} - y_i)^2 \\
&= 2 \sum_{i=1}^{n} (a_1 x_{i1} + a_2 x_{i2} + \cdots + a_m x_{im} - y_i) x_{im} = 0
\end{align*}
\]

or

\[
\frac{\partial E(a_j)}{\partial a_j} = 2 \sum_{i=1}^{n} (a_1 x_{i1} + a_2 x_{i2} + \cdots + a_m x_{im} - y_i) x_{ij} = 0
\]

Rearranging these normal equations gives
\[ a_1 \sum_{i=1}^{n} x_{i1}^2 + a_2 \sum_{i=1}^{n} x_{i2}^2 x_{i1} + \cdots + a_m \sum_{i=1}^{n} x_{im} x_{i1} = \sum_{i=1}^{n} y_i x_{i1} \]
\[ a_1 \sum_{i=1}^{n} x_{i1} x_{i2} + a_2 \sum_{i=1}^{n} x_{i2}^2 + \cdots + a_m \sum_{i=1}^{n} x_{im} x_{i2} = \sum_{i=1}^{n} y_i x_{i2} \]
\[ \vdots \]
\[ a_1 \sum_{i=1}^{n} x_{i1} x_{im} + a_2 \sum_{i=1}^{n} x_{i2} x_{im} + \cdots + a_m \sum_{i=1}^{n} x_{im}^2 = \sum_{i=1}^{n} y_i x_{im} \]

or (for each \( j \)):
\[ a_1 \sum_{i=1}^{n} x_{i1} x_{ij} + a_2 \sum_{i=1}^{n} x_{i2} x_{ij} + \cdots + a_m \sum_{i=1}^{n} x_{im} x_{ij} = \sum_{i=1}^{n} y_i x_{ij} \]

This provides a closed system of \( m \) normal equations, one for each coefficient, e.g.,
\[ \frac{\partial E(a)}{\partial a_j} = 0 \quad \text{for} \quad j=1, 2, \ldots, m. \]

In matrix form
\[
\begin{bmatrix}
\sum_{i=1}^{n} x_{i1}^2 & \sum_{i=1}^{n} x_{i2} x_{i1} & \cdots & \sum_{i=1}^{n} x_{im} x_{i1} \\
\sum_{i=1}^{n} x_{i1} x_{i2} & \sum_{i=1}^{n} x_{i2}^2 & \cdots & \sum_{i=1}^{n} x_{im} x_{i2} \\
\vdots & \vdots & \ddots & \vdots \\
\sum_{i=1}^{n} x_{i1} x_{im} & \sum_{i=1}^{n} x_{i2} x_{im} & \cdots & \sum_{i=1}^{n} x_{im}^2 \\
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_2 \\
\vdots \\
a_m \\
\end{bmatrix}
= 
\begin{bmatrix}
\sum_{i=1}^{n} y_i x_{i1} \\
\sum_{i=1}^{n} y_i x_{i2} \\
\vdots \\
\sum_{i=1}^{n} y_i x_{im} \\
\end{bmatrix}
\]

or simply
\[ N \cdot x = B \]

where \( N \) is the (known) coefficient matrix, \( x \) the vector with the unknowns \((a_j)\), and \( B \) contains weighted sums of known (observed) quantities. Solve for the \( a_j \) in the \( x \) vector (\( N \) is square and of full rank):
\[ N^{-1} \cdot N \cdot x = N^{-1} \cdot B \]
\[ x = N^{-1} \cdot B \]

where \( x \) is the solution for the \( a_j \). The resulting \( a_j \) values are the ones which satisfy (3.7) and therefore the same ones which, in combination with the chosen basis, produce the "best" fit to the \( n \) data points such that (3.6) is minimized.

Last time we found the solution to a general linear least squares problem led us to the matrix form
\[ \begin{bmatrix} \sum_{i=1}^{n} x_{i1}^2 & \sum_{i=1}^{n} x_{i2} x_{i1} & \cdots & \sum_{i=1}^{n} x_{in} x_{i1} \\ \sum_{i=1}^{n} x_{i1} x_{i2} & \sum_{i=1}^{n} x_{i2}^2 & \cdots & \sum_{i=1}^{n} x_{in} x_{i2} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} x_{i1} x_{im} & \sum_{i=1}^{n} x_{i2} x_{im} & \cdots & \sum_{i=1}^{n} x_{in} x_{im} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i x_{i1} \\ \sum_{i=1}^{n} y_i x_{i2} \\ \vdots \\ \sum_{i=1}^{n} y_i x_{im} \end{bmatrix} \]

or simply

\[ N \cdot x = B \]  

where \( N \) is the (known) coefficient matrix, \( x \) the vector with the unknowns \( (a_j) \), and \( B \) contains weighted sums of known (observed) quantities. Solve for the \( a_j \) in the \( x \) vector (\( N \) is square and of full rank):

\[ N^{-1} \cdot N \cdot x = N^{-1} \cdot B \]

\[ x = N^{-1} \cdot B \]  

where \( x \) is the solution for the \( a_j \). The resulting \( a_j \) values are the ones which satisfy (3.7) and therefore the same ones which, in combination with the chosen basis, produce the "best" fit to the \( n \) data points such that (3.6) is minimized.

We will now look at a simpler approach to the same problem using matrix algebra. We have

\[ \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \]

\[ A \cdot x = b \]

Since \( m < n \), the \( A \) matrix is rectangular. This can be written

\[ \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \]

\[ \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \]

\[ C \cdot X = e \]

These matrices describe the system listed earlier. We wish to find the \( a_j \) values which minimize \( E = e^T e \). The matrix form can be partitioned as

\[ \begin{bmatrix} A & -b \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} = e \]
\[
\begin{bmatrix}
A & -b
\end{bmatrix}
\begin{bmatrix}
x_1
\end{bmatrix}
= A \cdot x + (-b \cdot 1) = A \cdot x - b = e
\]

We will, for now, refer to the \([A -b]\) matrix as the \(C\) matrix and the \([x 1]^T\) column vector as the \(X\) vector. So, we have

\[
C \cdot X = e
\]

(recall \(C\) is an \(n \times m+1\) rectangular matrix). The \(i\)'th error, \(e_i\), is the dot product:

\[
C_i \cdot X = e_i = \begin{bmatrix}
1 & x_{i1} & x_{i2} & \cdots & x_{im} & y_i
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_2 \\
\vdots \\
a_m \\
1
\end{bmatrix} = \begin{bmatrix}
x_{i1} \\
x_{i2} \\
\vdots \\
x_{im} \\
y_i
\end{bmatrix}
\]

where \(C_i\) is the \(i\)'th row vector in \(C\). The squared \(i\)'th error, \(e_i^2\), is then

\[
e_i^T \cdot e_i = \begin{bmatrix}
a_1 & a_2 & \cdots & a_m & 1
\end{bmatrix}
\begin{bmatrix}
x_{i1} \\
x_{i2} \\
\vdots \\
x_{im} \\
y_i
\end{bmatrix}
\]

where we have used the reversal rule for transposed products. The sum of the \(e_i^2\) over all the \(i\)'s is thus

\[
e^T \cdot e = X^T \cdot C^T \cdot C \cdot X =
\]

The product \(C^T C\) can be computed to form a new matrix \(R\). Since \(C^T_{m+1,n} C_{n,m+1} = R_{m+1,m+1}\) the resulting \(R\) matrix is square and symmetric. So,

\[
E = e^T \cdot e = X^T \cdot C^T \cdot C \cdot X = X^T \cdot R \cdot X
\]

To minimize \(E\), we find

\[
\frac{\partial E}{\partial a_j} = \frac{\partial}{\partial a_j} (X^T \cdot R \cdot X) = \frac{\partial X^T}{\partial a_j} (R \cdot X) + X^T \cdot R \cdot \frac{\partial X}{\partial a_j}
\]

For the 2nd coefficient (as an example), we get
\[
\frac{\partial X^T}{\partial a_2} = \begin{bmatrix}
0 & 1 & 0 & \cdots & 0 & 0
\end{bmatrix} = \left(\frac{\partial X}{\partial a_2}\right)^T
\]

Thus, the partial derivative of the error is
\[
\frac{\partial E}{\partial a_2} = \begin{bmatrix}
0 & 1 & 0 & \cdots & 0 & 0
\end{bmatrix} R \begin{bmatrix}
a_1 \\
a_2 \\
\vdots \\
a_m \\
1
\end{bmatrix} + \begin{bmatrix}
a_1 & a_2 & \cdots & a_m & 1
\end{bmatrix} R \begin{bmatrix}
0 \\
1 \\
\vdots \\
0 \\
0
\end{bmatrix}
\]

For all coefficients we set all the partial derivatives to zero:
\[
\frac{\partial E(a_j)}{\partial a_j} = 0 = \dot{X}^T \cdot R \cdot X + R \cdot \dot{X}
\]

where
\[
\dot{X}^T = \frac{\partial X^T}{\partial a_j} = \begin{bmatrix}
1 & 0 & \cdots & 0 & 0 \\
0 & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 1 & 0
\end{bmatrix} = \left[ I_m | O_m \right]
\]

where \(I_m\) is the \(m \times m\) identity matrix and \(O_m\) the null vector of length \(m\). Consider first \(R\) (= \(C^T C\):

\[
C = \begin{bmatrix}
x_{11} & x_{12} & \cdots & x_{1m} & -y_1 \\
x_{21} & x_{22} & \cdots & x_{2m} & -y_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
x_{n1} & x_{n2} & \cdots & x_{nm} & -y_n
\end{bmatrix}
\]

\[
C^T = \begin{bmatrix}
x_{11} & x_{21} & \cdots & x_{n1} & -y_1 \\
x_{12} & x_{22} & \cdots & x_{n2} & -y_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
x_{1m} & x_{2m} & \cdots & x_{nm} & -y_n
\end{bmatrix}
\]

\[
R = \begin{bmatrix}
\sum x_{i1}^2 & \sum x_{i2} x_{i1} & \cdots & \sum x_{im} x_{i1} & -\sum y x_{i1} \\
\sum x_{i2} x_{i1} & \sum x_{i2}^2 & \cdots & \sum x_{im} x_{i2} & -\sum y x_{i2} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\sum x_{im} x_{i1} & \sum x_{im} x_{i2} & \cdots & \sum x_{im}^2 & -\sum y x_{im} \\
\sum y x_{i1} & \sum y x_{i2} & \cdots & \sum y x_{im} & \sum y_i^2
\end{bmatrix}
\]

The \(R\) matrix should look familiar. Consider the partitioned matrix multiplication:
\[ \mathbf{R} = \mathbf{C}^T \mathbf{C} = \begin{bmatrix} \mathbf{A}^T & \mathbf{b}^T \\ \mathbf{b}^T & \mathbf{b}^T \end{bmatrix} \]

Notice that \( \mathbf{A}^T \mathbf{A} \) is matrix \( \mathbf{N} \) of the normal equations and \( -\mathbf{A}^T \mathbf{b} = (-\mathbf{b}^T \mathbf{A})^T \) equal matrix \( \mathbf{B} \). Because \( \mathbf{R} \) is symmetrical (so \( \mathbf{R} = \mathbf{R}^T \)) we have

\[ \mathbf{X}^T \cdot \mathbf{R} = \left( \mathbf{R} \cdot \mathbf{X} \right)^T \]

So

\[ \frac{\partial E(a_j)}{\partial a_j} = \dot{\mathbf{X}}^T \cdot \mathbf{R} \cdot \mathbf{X} + \ddot{\mathbf{X}}^T \cdot \mathbf{R} \cdot \mathbf{X} = 0 \]

\[ = 2 \dot{\mathbf{X}}^T \cdot \mathbf{R} \cdot \mathbf{X} = 0 \Rightarrow \dot{\mathbf{X}}^T \cdot \mathbf{R} \cdot \mathbf{X} = 0 \]

Consider \( \dot{\mathbf{X}}^T \cdot \mathbf{R} \):
Always multiplied by zero

And \( (X^T \cdot R) \cdot X \):

\[
\begin{bmatrix}
X^T \\
R
\end{bmatrix}
\begin{bmatrix}
I_{(m \times m)} \\
0
\end{bmatrix}
= 0
\]

Finally, since

\[
X^T \cdot R \cdot X = A^T \cdot A \cdot x - A^T \cdot b = 0
\]

\[
A^T \cdot A \cdot x = A^T \cdot b
\]

\[
[A^T \cdot A]^{-1} A^T \cdot A \cdot x = [A^T \cdot A]^{-1} A^T \cdot b
\]

\[
x = [A^T \cdot A]^{-1} A^T \cdot b
\]

or

\[
x = N^{-1} \cdot B
\]
as before. Therefore, the unknown values of the \( a_j \) (in the \( x \) vector) can be solved for directly from the system:

\[
\begin{align*}
\cdot x_{11} + a_2 x_{12} + \cdots + a_m x_{1m} &= y_1 \\
\cdot x_{21} + a_2 x_{22} + \cdots + a_m x_{2m} &= y_2 \\
\vdots \\
a_1 x_{n1} + a_2 x_{n2} + \cdots + a_m x_{nm} &= y_n
\end{align*}
\]

or simply

\[
A \cdot x = b
\]

where \( A \) is of order \( n \times m \) with \( n > m \). The least squares solution then becomes

\[
x = (A^T \cdot A)^{-1} A^T \cdot b
\]

where \( A^T A \) is a square matrix of full rank, with order \( r = m \), and thus invertable.

### Fitting a straight line, revisited

We will again consider the best-fitting line problem, this time with errors \( \sigma_i \) in the \( y \)-values. We want to measure how well the model agrees with the data, and for this purpose will use the \( \chi^2 \) function, i.e.

\[
\chi^2(a,b) = \sum_{i=1}^{n} \left( \frac{y_i - a - bx_i}{\sigma_i} \right)^2
\]

Minimizing \( \chi^2 \) will give the best weighted least squares solution. Again we set the partial derivatives to 0:

\[
\frac{\partial \chi^2}{\partial a} = 0 = -2 \sum_{i=1}^{n} \left( \frac{y_i - a - bx_i}{\sigma_i^2} \right)
\]

\[
\frac{\partial \chi^2}{\partial b} = 0 = -2 \sum_{i=1}^{n} \left( \frac{y_i - a - bx_i}{\sigma_i^2} \right) x_i
\]

Let us define the following:

\[
S = \sum \frac{1}{\sigma_i^2} \quad S_x = \sum \frac{x_i}{\sigma_i^2} \quad S_y = \sum \frac{y_i}{\sigma_i^2} \quad S_{xx} = \sum \frac{x_i^2}{\sigma_i^2} \quad S_{xy} = \sum \frac{x_i y_i}{\sigma_i^2}
\]

Then (3.12) reduces to

\[
a S + b S_x = S_y \\
a S_x + b S_{xx} = S_{xy}
\]

With

\[
\Delta = SS_{xx} - S_x^2
\]

we find
\[
\begin{align*}
    a &= \frac{S_{xx}S_y - S_xS_y}{\Delta} \\
    b &= \frac{SS_{xy} - S_xS_y}{\Delta}
\end{align*}
\]  

(3.13)

All this is swell but we must also estimate the uncertainties in \(a\) and \(b\). For the same \(\sigma_i\) we may get large differences in errors in \(a\) and \(b\). Although not shown here, consideration of propagation of errors shows that the variance \(\sigma^2\) in the value of any function is

\[
\sigma^2_j = \sum \sigma_i^2 \left( \frac{\partial f}{\partial y_i} \right)^2
\]

(3.14)

For our line we can directly find the derivatives of \(a\) and \(b\) with respect to \(y_i\) from (3.13):

\[
\frac{\partial a}{\partial y_i} = \frac{S_{xx} - S_x x_i}{\sigma_i^2 \Delta}
\]

\[
\frac{\partial b}{\partial y_i} = \frac{S x_i - S_x}{\sigma_i^2 \Delta}
\]

Inserting this result into (14.7) then gives

\[
\sigma^2_a = \sum \sigma_i^2 \left[ \frac{S_{xx} - S_x x_i}{\sigma_i^2 \Delta} \right]^2 = \sum \frac{S_{xx}^2 - 2S_{xx}S_x x_i + S_x x_i^2}{\sigma_i^2 \Delta^2}
\]

\[
= \frac{S_{xx}^2}{\Delta^2} \sum \frac{1}{\sigma_i^2} - 2 \frac{S_{xx}}{\Delta^2} \sum \frac{x_i}{\sigma_i^2} + \frac{S_x^2}{\Delta^2} \sum \frac{x_i^2}{\sigma_i^2} = \frac{S_{xx}^2}{\Delta^2} - \frac{2S_{xx}S_x}{\Delta^2} + \frac{S_x^2}{\Delta^2}
\]

\[
= \frac{S_{xx}(S_{xx} - S_x^2)}{\Delta^2} = \frac{S_{xx}}{\Delta}
\]

(3.15)

and

\[
\sigma^2_b = \sum \sigma_i^2 \left[ \frac{S x_i - S_x}{\sigma_i^2 \Delta} \right]^2 = \sum \frac{S^2 x_i^2 - 2SS x_i + S_x^2}{\sigma_i^2 \Delta^2}
\]

Figure 3-9. The uncertainty in the line fit depends to a large extent on the distribution of the x-positions.
\[ \frac{S^2}{\Delta} \sum \frac{x_i^2}{\sigma_i^2} - \frac{2SS}{\Delta} \sum \frac{x_i}{\sigma_i^2} + \frac{S^2}{\Delta} \sum \frac{1}{\sigma_i^2} = \frac{S^2}{\Delta} \sum \frac{1}{\sigma_i^2} - \frac{2SS^2}{\Delta^2} + \frac{S^2}{\Delta^2} \]

\[ = \frac{S(S_x^2 - S^2)}{\Delta^2} = \frac{S}{\Delta} \]

Similarly, we can find the covariance \( \sigma_{ab} \) from

\[ \sigma_{ab}^2 = \sum \sigma_i^2 \left( \frac{\partial a}{\partial y_i} \right) \left( \frac{\partial b}{\partial y_i} \right) = -\frac{S}{\Delta} \]

Then, the correlation coefficient becomes

\[ r = -\frac{S}{\sqrt{SS_{xx}}} \]

(3.17)

It is therefore useful to shift the origin to \( \mathbf{x} \) where \( r = 0 \).

Finally, we must check if the fit is meaningful. We use the \( \chi^2 \) value computed and get critical \( \chi^2_{\alpha} \) for \( n - 2 \) degrees of freedom which we use to see if \( \chi^2 \) exceeds this value. If it doesn't we may say the fit is significant at the \( \alpha \) level.

What if some data constraints are more reliable than others? We may give that residual more weight than the others:

\[ e = \begin{bmatrix} e_1 \\ 2e_2 \\ \vdots \\ e_n \end{bmatrix} \]

In general, we can use weights \( w_i \) for each error so that the new error \( e_i' = e_i \cdot w_i \). We do this by introducing a weight matrix \( w \) which is a diagonal matrix:

\[ w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \]

Now the sum of the squared errors, \( E \), becomes

\[ E = e^T \cdot e = e^T \cdot w^T \cdot w \cdot e = e^T \cdot W \cdot e \]

where we have introduced \( W = w^T \cdot w \). Since we have \( w^T \cdot e = w(A^T \cdot x - b) \) we obtain

\[ E = (w^T A \cdot x - w^T b)(w^T A \cdot x - w^T b) = (x^T A^T w^T - b^T w^T)(w^T w \cdot A \cdot x - w^T b) \]

\[ = x^T A^T w \cdot w^T A \cdot x - x^T A^T w \cdot b - b^T w \cdot A \cdot x + b^T w \cdot b \]

We substitute for \( W \). Then
\[
\frac{\partial E}{\partial a_j} = 0 = x^T A^T W A x + x^T A^T W A \hat{x} - x^T A^T W b - b^T W A \hat{x}
\]

Since \( x \) only contains the \( a_j \), we have \( x^T = \hat{x} = I \). We find

\[
A^T W A x + x^T A^T W A - A^T W b - b^T W A = 0
\]

Again, the 2nd and 4th term are the transpose of the 1st and 3rd. Because each term represent a symmetrical matrix our equation reduces to

\[
2A^T W A x - 2A^T W b = 0
\]

which gives us the weighted linear least squares solution

\[
x = \left[ A^T W A \right]^{-1} A^T W b
\]

(3.18)