

A Don's Vade Mecum

R.G. Grainger

May 10, 2017

Contents

1	Maths	2
1.1	Delta functions	2
1.2	Expansions	2
1.3	Changing the variable of integration	2
1.4	Matrices	3
1.5	Vector Operation	3
1.5.1	Cartesian Coordinates	3
1.5.2	Spherical Coordinates	3
1.6	Vector Identities	4
1.7	Spherical Coordinates	4
2	Computing	5
2.1	Setting an Xterm up nicely	5
2.2	LATEX	5
2.2.1	How to display a \	5
2.2.2	Framing a block of text	5
2.2.3	Stopping a float floating	5
2.3	LINUX	5
2.3.1	Creating a symbolic logical link	5
2.3.2	Extracting the files from a tar file	6
2.3.3	Seeing how much space a directory and its subdirectories are taking	6
2.4	HTML	6
2.5	WINDOWS	6
2.5.1	How to login as admin	6
2.5.2	How to see event logs	6
2.5.3	How to eliminate the “Open File Security Warning” from programmes accessed from the file server.	6
2.6	IDL	7
2.6.1	Array Index Order	7
2.6.2	Confusion	7
2.6.3	Matrix Multiplication	7
2.6.4	Example	8
2.6.5	Optimisation	8
2.6.6	Forming an Array by Repeating a Vector	9
2.6.7	Using Arrays to Improve Efficiency	9
2.7	Microsoft Word	11
2.7.1	How to Update Cross-References	11
2.8	Adobe Acrobat	11

2.8.1	How to crop in Acrobat X	11
2.9	Account Administration	11
2.9.1	How to Update EODG List	11
3	Mathematical typesetting	12
3.1	Distinguishing elements in vectors and matrices	12

1 Maths

1.1 Delta functions

The *Dirac* delta function $\delta(x)$ has the value 0 everywhere except at $x = 0$ its value is infinitely large in such a way that its total integral is 1.

The *Kronecker* delta function $\delta_{i,j}$ is defined as

$$\delta_{i,j} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

1.2 Expansions

$$\text{Binomial } (1+x)^n = 1 + nx + \frac{n(n-1)}{2!}x^2 + \frac{n(n-1)(n-2)}{3!}x^3 + \dots$$

1.3 Changing the variable of integration

Tip replace one variable with another use

$$I = \int_a^b f(g(x)) \frac{dg}{dx} dx = \int_{g(a)}^{g(b)} f(y) dy$$

$$I = \int_a^b y dy = \int_a^b y \frac{dy}{dx} dx$$

More generally suppose x, y are given as functions of two new variables s and t . Then

$$dx dy = |J| ds dt$$

where J is the Jacobian of x and y with respect to s and t . In this case

$$|J| = \begin{vmatrix} \frac{\partial x}{\partial s} & \frac{\partial x}{\partial t} \\ \frac{\partial y}{\partial s} & \frac{\partial y}{\partial t} \end{vmatrix}$$

Example. For a polar coordinate system

$$x = r \cos \phi \tag{1}$$

$$y = r \sin \phi \tag{2}$$

so

$$|J| = \begin{vmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \phi} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \phi} \end{vmatrix} = \begin{vmatrix} \cos \phi & -r \sin \phi \\ \sin \phi & r \cos \phi \end{vmatrix} = r \cos^2 \phi - (-r \sin^2 \phi) = r$$

So $dx dy = r d\phi dr$.

1.4 Matrices

Determinant of a 2×2 matrix

$$\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \det \mathbf{A} = |\mathbf{A}| = ad - bc \quad (3)$$

Determinant of a 3×3 matrix

$$\mathbf{A} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}, \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a(ei - fh) - b(di - fg) + c(dh - eg) \quad (4)$$

1.5 Vector Operation

1.5.1 Cartesian Coordinates

Let \mathbf{r} be an arbitrary vector then

Del, Grad ∇

$$\nabla \mathbf{r} = \frac{\partial \mathbf{r}}{\partial x} \hat{\mathbf{x}} + \frac{\partial \mathbf{r}}{\partial y} \hat{\mathbf{y}} + \frac{\partial \mathbf{r}}{\partial z} \hat{\mathbf{z}} \quad (5)$$

Div, $\nabla \cdot$

$$\nabla \cdot \mathbf{r} = \frac{\partial r_x}{\partial x} + \frac{\partial r_y}{\partial y} + \frac{\partial r_z}{\partial z} \quad (6)$$

Curl, $\nabla \times$

$$\nabla \times \mathbf{r} = \begin{vmatrix} \hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{z}} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ r_x & r_y & r_z \end{vmatrix} = \left(\frac{\partial r_z}{\partial y} - \frac{\partial r_y}{\partial z} \right) \hat{\mathbf{x}} - \left(\frac{\partial r_z}{\partial x} - \frac{\partial r_x}{\partial z} \right) \hat{\mathbf{y}} + \left(\frac{\partial r_y}{\partial x} - \frac{\partial r_x}{\partial y} \right) \hat{\mathbf{z}} \quad (7)$$

Laplacian, ∇^2

$$\nabla^2 \mathbf{r} = \nabla \cdot \nabla \mathbf{r} = \frac{\partial^2 \mathbf{r}}{\partial x^2} + \frac{\partial^2 \mathbf{r}}{\partial y^2} + \frac{\partial^2 \mathbf{r}}{\partial z^2} \quad (8)$$

1.5.2 Spherical Coordinates

Let \mathbf{f} be an arbitrary vector then

Del, Grad ∇

$$\nabla \mathbf{f} = \frac{\partial \mathbf{f}}{\partial r} \hat{\mathbf{r}} + \frac{1}{r} \frac{\partial \mathbf{f}}{\partial \theta} \hat{\boldsymbol{\theta}} + \frac{1}{r \sin \theta} \frac{\partial \mathbf{f}}{\partial \phi} \hat{\boldsymbol{\phi}} \quad (9)$$

Div, $\nabla \cdot$

$$\nabla \cdot \mathbf{f} = \frac{1}{r^2} \frac{\partial (r^2 f_r)}{\partial r} + \frac{1}{r \sin \theta} \frac{\partial (\sin \theta f_\theta)}{\partial \theta} + \frac{1}{r \sin \theta} \frac{\partial f_\phi}{\partial \phi} \quad (10)$$

Laplacian, ∇^2

$$\nabla^2 \mathbf{f} = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial \mathbf{f}}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial \mathbf{f}}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 \mathbf{f}}{\partial \phi^2} \quad (11)$$

1.6 Vector Identities

$$\nabla \times (\nabla \times \mathbf{r}) = \nabla(\nabla \cdot \mathbf{r}) - \nabla^2(\mathbf{r}) \quad (12)$$

Proof

$$\nabla \times (\nabla \times \mathbf{r}) = \left(\frac{\partial}{\partial y} \left(\frac{\partial r_y}{\partial x} - \frac{\partial r_x}{\partial y} \right) - \frac{\partial}{\partial z} \left(- \left(\frac{\partial r_z}{\partial x} - \frac{\partial r_x}{\partial z} \right) \right) \right) \mathbf{i} \quad (13)$$

$$- \left(\frac{\partial}{\partial x} \left(\frac{\partial r_y}{\partial x} - \frac{\partial r_x}{\partial y} \right) - \frac{\partial}{\partial z} \left(\frac{\partial r_z}{\partial y} - \frac{\partial r_y}{\partial z} \right) \right) \mathbf{j} \quad (14)$$

$$+ \left(\frac{\partial}{\partial x} \left(- \left(\frac{\partial r_z}{\partial x} - \frac{\partial r_x}{\partial z} \right) \right) - \frac{\partial}{\partial y} \left(\frac{\partial r_z}{\partial y} - \frac{\partial r_y}{\partial z} \right) \right) \mathbf{k} \quad (15)$$

$$= \left(\frac{\partial^2 r_y}{\partial y \partial x} - \frac{\partial^2 r_x}{\partial y^2} + \frac{\partial^2 r_z}{\partial z \partial x} - \frac{\partial^2 r_x}{\partial z^2} \right) \mathbf{i} \quad (16)$$

$$- \left(\frac{\partial}{\partial x} \left(\frac{\partial r_y}{\partial x} - \frac{\partial r_x}{\partial y} \right) - \frac{\partial}{\partial z} \left(\frac{\partial r_z}{\partial y} - \frac{\partial r_y}{\partial z} \right) \right) \mathbf{j} \quad (17)$$

$$+ \left(\frac{\partial}{\partial x} \left(- \left(\frac{\partial r_z}{\partial x} - \frac{\partial r_x}{\partial z} \right) \right) - \frac{\partial}{\partial y} \left(\frac{\partial r_z}{\partial y} - \frac{\partial r_y}{\partial z} \right) \right) \mathbf{k} \quad (18)$$

still to do lines 11 and 12.

$$\nabla \cdot (a\mathbf{r}) = \mathbf{r} \cdot \nabla a + a\nabla \cdot \mathbf{r} \quad (19)$$

Proof

$$\mathbf{r} \cdot \nabla a = \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix} \cdot \begin{pmatrix} \frac{\partial a}{\partial x} \\ \frac{\partial a}{\partial y} \\ \frac{\partial a}{\partial z} \end{pmatrix} = r_x \frac{\partial a}{\partial x} + r_y \frac{\partial a}{\partial y} + r_z \frac{\partial a}{\partial z} \quad (20)$$

$$a\nabla \cdot \mathbf{r} = a \left(\frac{\partial r_x}{\partial x} + \frac{\partial r_y}{\partial y} + \frac{\partial r_z}{\partial z} \right) \nabla \cdot (a\mathbf{r}) \quad (21)$$

$$= \frac{\partial a r_x}{\partial x} + \frac{\partial a r_y}{\partial y} + \frac{\partial a r_z}{\partial z} \quad (22)$$

$$= \frac{\partial a}{\partial x} r_x + a \frac{\partial r_x}{\partial x} + \frac{\partial a}{\partial y} r_y + a \frac{\partial r_y}{\partial y} + \frac{\partial a}{\partial z} r_z + a \frac{\partial r_z}{\partial z} \quad (23)$$

$$= \underbrace{r_x \frac{\partial a}{\partial x} + r_y \frac{\partial a}{\partial y} + r_z \frac{\partial a}{\partial z}}_{\mathbf{r} \cdot \nabla a} + \underbrace{a \left(\frac{\partial r_x}{\partial x} + \frac{\partial r_y}{\partial y} + \frac{\partial r_z}{\partial z} \right)}_{a\nabla \cdot \mathbf{r}} \quad (24)$$

Also

$$\begin{aligned} \mathbf{u} \cdot \mathbf{v} &= \mathbf{v} \cdot \mathbf{u} \\ \frac{d}{dt}(\mathbf{u} \cdot \mathbf{v}) &= \frac{d\mathbf{u}}{dt} \cdot \mathbf{v} + \mathbf{u} \cdot \frac{d\mathbf{v}}{dt} \\ \frac{d}{dt} \left(\frac{\mathbf{u} \cdot \mathbf{u}}{2} \right) &= \mathbf{u} \cdot \frac{d\mathbf{u}}{dt} = \frac{d\mathbf{u}}{dt} \cdot \mathbf{u} \end{aligned}$$

1.7 Spherical Coordinates

Spherical polar coordinates (r, θ, ϕ) are defined by

$$x = r \sin \theta \cos \phi \quad (25)$$

$$y = r \sin \theta \sin \phi \quad (26)$$

$$z = r \cos \theta \quad (27)$$

The counter transform from Cartesian to spherical polar coordinates is

$$r = \sqrt{x^2 + y^2 + z^2} \quad (28)$$

$$\theta = \cos^{-1} \frac{z}{\sqrt{x^2 + y^2 + z^2}} \quad (29)$$

$$\phi = \tan^{-1} \frac{y}{x} \quad (30)$$

2 Computing

2.1 Setting an Xterm up nicely

```
/usr/X11R6/bin/xterm -sb -ls -fn arial -display @d&@;
```

2.2 LATEX

One way to change the directories in which LaTeX looks for its input files is by setting the environment variable TEXINPUTS.

right-click my computer and select properties:advanced:environmental variables

create or edit TEXINPUTS to make:

TEXINPUTS '.:\$HOME/myown:'

Setting TEXINPUTS in this way will causes LaTeX to look for files first in the current directory, then in the subdirectory myown of your home directory, and then finally in the standard system directories.

2.2.1 How to display a \

Use `\textbackslash`

2.2.2 Framing a block of text

If you need to draw a box around some tex then use

```
\fbox{\parbox{\linewidth}{Text to Frame}}
```

2.2.3 Stopping a float floating

The caption package can be used to prevent floating and to give all the other features of the figure or table environments. It provides the command `\captionof` that takes a counter (figure or table) as argument and of course the caption text, for example:

```
\usepackage{caption}
...
\begin{center}
\includegraphics{filename}%
\captionof{figure}{text}\label{labelname}%
\end{center}
```

2.3 LINUX

2.3.1 Creating a symbolic logical link

```
ln -s existingfile newname
```

2.3.2 Extracting the files from a tar file

```
tar -xvfv myfile.tar
```

In the above example command the system would uncompress (untar) the myfile.tar file in the current directory.

```
tar -xvzvf myfile.tar.gz
```

In the above example command the system would uncompress (untar) the myfile.tar.gz file in the current directory. Note that .tar.gz can be abbreviated as .tgz so

```
tar -xvzvf myfile.tgz
```

Note: There is no “untar” linux / unix command.

2.3.3 Seeing how much space a directory and its subdirectories are taking

```
du -chs directory
```

The 'b' options is vital as otherwise the way the file system works influences the result.

2.4 HTML

```
<element class="classname">
```

2.5 WINDOWS

2.5.1 How to login as admin

The built-in Administrator account is hidden from Welcome Screen when a user account with Administrator privileges exists and enabled. In Windows XP Professional, press CTRL + ALT + DEL twice at the Welcome Screen and input your Administrator password in the classic logon window that appears.

2.5.2 How to see event logs

Open the Start Menu, type eventvwr.msc, and press Enter.

2.5.3 How to eliminate the “Open File Security Warning” from programmes accessed from the file server.

Open the Control Panel

1. Open Internet Options
2. Click the Security Tab
3. Click on Local Intranet
4. Click on Sties
5. Click Advanced
6. Type the drive letter of your file server where the application is located in the “Add this website to this zone” box. e.g. H:
7. Click Add
8. Click Close
9. Click OK

10. Close Internet Options by clicking OK

11. Close the Control Panel

2.6 IDL

2.6.1 Array Index Order

In the standard mathematical notation an $m \times n$ array has m rows and n columns. The first dimension of an array (or matrix) is the row, and the second dimension is the column. For example,

$$\begin{bmatrix} A_{0,0} & A_{0,1} & A_{0,2} & \dots & A_{0,n-1} \\ A_{1,0} & A_{1,1} & A_{1,2} & \dots & A_{1,n-1} \\ A_{2,0} & A_{2,1} & A_{2,2} & \dots & A_{2,n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ A_{m-1,0} & A_{m-1,1} & A_{m-1,2} & \dots & A_{m-1,n-1} \end{bmatrix}$$

Computer memory is sequential, memory addresses start at zero and increase serially to the highest available location. Multidimensional arrays are formed by software that maps the elements of a multi-dimensional array into a contiguous linear span of memory. There are two ways that such an array can be represented in one-dimensional linear memory, *row major* and *column major*. For example the array

$$\begin{bmatrix} A_{0,0} & A_{0,1} \\ A_{1,0} & A_{1,1} \end{bmatrix}$$

is stored:

Contiguous First Dimension (Column Major) $A_{0,0}, A_{1,0}, A_{0,1}, A_{1,1}$

Contiguous Second Dimension (Row Major) $A_{0,0}, A_{0,1}, A_{1,0}, A_{1,1}$

The Interactive Data Language (IDL) is similar to FORTRAN in using column major format to order arrays.¹

2.6.2 Confusion

In image processing, the convention is that the first dimension of an image array is the column, and the second dimension is the row. Hence, the dominant convention in IDL documentation is to refer to the first dimension of an array as the column and the second dimension as the row. So if we print a 2×3 array instead of formatting the output as two rows and three columns IDL prints three rows and two columns. To avoid any confusion here I will use dimension 1 and dimension 2.

2.6.3 Matrix Multiplication

IDL provides two operators for matrix multiplication. The $\#$ operator which computes the product array elements by multiplying dimension 2 of the first array by dimension 1 of the second array. This is illustrated in Figure 1. Using this operator an $m \times n$ matrix times a $n \times p$ matrix gives a $m \times p$ matrix.

¹C and Pascal use row major format.

$$\begin{pmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{pmatrix} \#\# \begin{pmatrix} \mathbf{y} & \mathbf{y} \\ \mathbf{y} & \mathbf{y} \\ \mathbf{y} & \mathbf{y} \\ \mathbf{y} & \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{z} & \mathbf{z} \\ \mathbf{z} & \mathbf{z} \\ \mathbf{z} & \mathbf{z} \end{pmatrix}$$

Figure 1: Conventional matrix multiplication (dimension 2 of the first array by dimension 1 of the second array).

The `##` operator computes the product array elements by multiplying dimension 1 of the first array by dimension 2 of the second array. This is shown in Figure 2. Using this operator an $n \times m$ matrix times a $p \times n$ matrix gives a $p \times m$ matrix.

$$\begin{pmatrix} \mathbf{y} & \mathbf{y} \\ \mathbf{y} & \mathbf{y} \\ \mathbf{y} & \mathbf{y} \\ \mathbf{y} & \mathbf{y} \\ \mathbf{y} & \mathbf{y} \end{pmatrix} \# \begin{pmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{z} & \mathbf{z} \\ \mathbf{z} & \mathbf{z} \\ \mathbf{z} & \mathbf{z} \end{pmatrix}$$

Figure 2: Reverse matrix multiplication (dimension 1 of the first array by dimension 2 of the second array).

Hence the conventional matrix multiplication \mathbf{AB} is coded in IDL as either $\mathbf{A}\#\mathbf{B}$ or $\mathbf{B}\#\#\mathbf{A}$. The resultant arrays are identical.

2.6.4 Example

Let's imagine we have a vector function $\mathbf{y} = \mathbf{f}(\mathbf{x})$ and we want to transform a covariance matrix \mathbf{S}_x to \mathbf{S}_y using the Jacobian, \mathbf{K} . \mathbf{S}_x has dimension $n \times n$ and \mathbf{S}_y has dimension $m \times m$. The transform is expressed mathematically as

$$\mathbf{S}_y = \mathbf{K}\mathbf{S}_x\mathbf{K}^T$$

Remember the Jacobian of the matrix of derivatives of \mathbf{f} with respect to \mathbf{x} i.e.

$$K_{ij} = \frac{\partial f_i}{\partial x_j} \quad \text{or} \quad \mathbf{K} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \frac{\partial f_1(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial f_2(\mathbf{x})}{\partial x_1} & \frac{\partial f_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_2(\mathbf{x})}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \frac{\partial f_m(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_m(\mathbf{x})}{\partial x_n} \end{bmatrix}.$$

The dimensions of \mathbf{K} are $n \times m$ (i.e. the size of \mathbf{x} times the number of functions).

2.6.5 Optimisation

For large arrays it is much quicker to work sequentially through the data in an array in the order in which it is stored. This minimises the amount of I/O if large arrays are being manipulated. So provided the speed of 'Function' is independent of the values of 'I' and 'J' it is always better to do

```

For J = 1, 10000 do begin
  For I = 1, 10000 do begin
    A(I, J) = Function(I, J)
  EndFor
EndFor

```

than it is to do

```

For I = 1, 10000 do begin
  For J = 1, 10000 do begin
    A(I, J) = Function(I, J)
  EndFor
EndFor

```

2.6.6 Forming an Array by Repeating a Vector

Given a vector $[Z(0), Z(1), \dots, Z(M-1)]$ it is not uncommon to want to create an array which whose columns or rows are composed of Z repeated N times. The array

$$ZARRAY = \begin{pmatrix} Z(0) & Z(0) & Z(0) & \dots & Z(0) \\ Z(1) & Z(1) & Z(1) & \dots & Z(1) \\ Z(2) & Z(2) & Z(2) & \dots & Z(2) \\ \vdots & \vdots & \vdots & & \vdots \\ Z(M-1) & Z(M-1) & Z(M-1) & \dots & Z(M-1) \end{pmatrix}$$

where the vector Z is repeated N times can be created by the IDL statement

```
ZARRAY = Z ## Replicate(1, N)
```

or

```
ZARRAY = Replicate(1, N) # Z
```

Alternatively the array

$$ZARRAY = \begin{pmatrix} Z(0) & Z(1) & Z(2) & \dots & Z(M-1) \\ Z(0) & Z(1) & Z(2) & \dots & Z(M-1) \\ Z(0) & Z(1) & Z(2) & \dots & Z(M-1) \\ \vdots & \vdots & \vdots & & \vdots \\ Z(0) & Z(1) & Z(2) & \dots & Z(M-1) \end{pmatrix}$$

where the vector Z is repeated N times can be created by the IDL statement

```
ZARRAY = Z # Replicate(1, N)
```

or

```
ZARRAY = Replicate(1, N) ## Z
```

2.6.7 Using Arrays to Improve Efficiency

It is not uncommon to want to do something like this

```

Sum = Fltarr(N)
For I = 0, N-1 do begin
  Sum(I) = 0
  For J = 0, M-1 do begin
    Sum(I) = Sum(I) + A(J) * Z(I)
  EndFor
EndFor

```

How do we rewrite this efficiently? The easiest substitution is to replace the second summation by a implied loop over J, i.e.

```

Sum = Fltarr(N)
For I = 0, N-1 do begin
    Sum(I) = Total(A* Z(I))
EndFor

```

However this still leads us with an explicit summation. A better way is to use array multiplication, e.g.

$$\begin{pmatrix} \text{Sum}(0) \\ \text{Sum}(1) \\ \text{Sum}(2) \\ \vdots \\ \text{Sum}(N-1) \end{pmatrix} = \begin{pmatrix} A(0) & A(1) & A(2) & \dots & A(M-1) \end{pmatrix} \underbrace{\begin{pmatrix} Z(0) & Z(1) & Z(2) & \dots & Z(N-1) \\ Z(0) & Z(1) & Z(2) & \dots & Z(N-1) \\ Z(0) & Z(1) & Z(2) & \dots & Z(N-1) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ Z(0) & Z(1) & Z(2) & \dots & Z(N-1) \end{pmatrix}}_{N \times M}$$

The IDL statement that implements this is

```
Sum = Z # replicate(1,M) # A
```

A more complicated case is to want to implement

```
Sum = Fltarr(N)
For I = 0, N-1 do begin
  Sum(I) = 0
  For J = 0, M-1 do begin
    Sum(I) = Sum(I) + cos(A(m) * Z(I))
  EndFor
EndFor
```

We cannot do this through matrix multiplication as there is a cosine operation before the addition so we use the matrix product, i.e.

$$\underbrace{\begin{pmatrix} A(0) & A(0) & A(0) & \dots & A(0) \\ A(1) & A(1) & A(1) & \dots & A(1) \\ A(2) & A(2) & A(2) & \dots & A(2) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ A(M-1) & A(M-1) & A(M-1) & \dots & A(M-1) \end{pmatrix}}_{N \times M} \times \underbrace{\begin{pmatrix} Z(0) & Z(1) & Z(2) & \dots & Z(N-1) \\ Z(0) & Z(1) & Z(2) & \dots & Z(N-1) \\ Z(0) & Z(1) & Z(2) & \dots & Z(N-1) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ Z(0) & Z(1) & Z(2) & \dots & Z(N-1) \end{pmatrix}}_{N \times M}$$

We want to take the cosine of the individual terms and then find the sum of the columns of the product. This is achieved by the IDL statement

```
Sum = Total(Cos(Replicate(1,N) # A * Z # (Replicate(1,M)), 2)
```

2.7 Microsoft Word

2.7.1 How to Update Cross-References

F9

2.8 Adobe Acrobat

2.8.1 How to crop in Acrobat X

Use CNTRL-SHIFT-T to bring up dialogue box.

2.9 Account Administration

2.9.1 How to Update EODG List

1. Use Remote desktop to access termserv.physics.ox.ac.uk
2. Start outlook
3. Open address book
4. Search for EODG
5. Edit

3 Mathematical typesetting

3.1 Distinguishing elements in vectors and matrices

x_i is the i^{th} element of \mathbf{x} .

\mathbf{x}_j is the j^{th} vector from an ensemble of \mathbf{x} vectors.

$x_{i,j}$ is the i^{th} element of the j^{th} vector.

$X_{i,j}$ is the element in the i^{th} row column and j^{th} of the matrix \mathbf{X} .