8 Matrix Representation of Transformations on Three-Dimensional Space

In chapter 4 we saw the need for transforming objects in two-dimensional space. When we draw three-dimensional pictures there will be many times when we need to make the equivalent linear transformations on three-dimensional space. As in the lower dimension, there are three basic types of transformation: translation, scaling and rotation. We will represent transformations as square matrices (now they will be 4 x 4). A general point in space relative to a fixed coordinate triad, the row vector (x, y, z), must be considered as a four-rowed column vector:

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

All the operations on matrices (addition, scalar multiple, transpose, premultiplication of a column vector and matrix product) that we saw in chapter 4 are easily extended to cope with 4 x 4 matrices and column vectors by simply changing the upper bound of the index ranges from 3 to 4. In this way we can generate a procedure ‘mult3’ (see listing 8.1) for multiplying two 4 x 4 matrices together. It is exactly equivalent to procedure ‘mult2’ in the two-dimensional case, and for the very same reasons. The procedure multiplies matrix A by matrix R to give matrix B, which is then copied into R. We also need the procedure ‘idR3’ (see listing 8.1) which sets R to the identity matrix.

Consider the case of a general linear transformation on points in three-dimensional space. A point (x, y, z) - ‘before’ - is transformed into (x',y', z') - ‘after’ - according to three *linear equations*:

$$x' = A_{11} \times x + A_{12} \times y + A_{13} \times z + A_{14}$$

$$y' = A_{21} \times x + A_{22} \times y + A_{23} \times z + A_{24}$$

$$z' = A_{31} \times x + A_{32} \times y + A_{33} \times z + A_{34}$$

and as usual we add the extra equation:

$$1 = A_{41} \times x + A_{42} \times y + A_{43} \times z + A_{44}$$

which if it is to be true for all *x*, *y* and *z* means that $A_{41} = A_{42} = A_{43} = 0$ and that $A_{44} = 1$

Then the equations may be written as a matrix equation where a column vector representing the 'after' point is the product of a matrix and the 'before' column vector:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{pmatrix} \times \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

So if we store the transformation as a matrix, we can transform every required point by considering it as a column vector and *premultiplying* it by a transformation matrix. As before, transformations may be combined simply by obeying the sequence of transformations in order. If their equivalent matrices are A, B, C, . . . , L, M, N, then the matrix equivalent to the combination is N × M × L × . . . × C × B × A. Remember the order. Since we are premultiplying a column vector, then the first transformation appears on the right of the matrix product and the last on the left.

As with the two-dimensional case, we note that the 'bottom row' of all transformation matrices is always (0, 0, 0, 1), and it is ofno real use in calculations. It is added only to form square matrices which are necessary for the formal definition of matrix multiplication. We may adjust this definition, and that of the multiplication of a matrix and a column vector, so that instead we use only the top three rows of the 4 X 4 matrices (in chapter 4 we used the top two rows of 3 x 3 matrices in listings 4.2a, 4.3a, 4.4a and 4.5a).

Listing 8.1

```
9100 REM mult3
9110 DEF PROCmult3
9120 LOCAL I%,J%,K%
9130 FOR I%=1 TO 3
9140 FOR J%=1 TO 4
9150 B(I%,J%)=A(I%,1)*R(1,J%)+A(I%,2)*R(2,J%)+A(I%,3)*R(3,J%)
9160 NEXT J%
9170 B(I%,4)=B(I%,4)+A(I%,4)
9180 NEXTI%
9190 FOR I%=1 TO 3
9200 FOR J%=1 TO 4
9210 R(I%,J%)=B(I%,J%)
9220 NEXT J%
9230 NEXT I%
9240 ENDPROC

9300 REM idR3
9310 DEF PROCidR3
9320 R(1,1)=1 : R(1,2)=0 : R(1,3)=0 : R(1,4)=0
9330 R(2,1)=0 : R(2,2)=1 : R(2,3)=0 : R(2,4)=0
9340 R(3,1)=0 : R(3,2)=0 : R(3,3)=1 : R(3,4)=0
9350 ENDPROC
```

Translation

Every point to be transformed is moved by a vector (TX, TY, TZ) say. This produces the following equations which relate to the ‘before’ and ‘after’ coordinates:

$$x' = 1 \times x + 0 \times y + 0 \times z + \mathrm{TX}$$

$$y' = 0 \times x + 1 \times y + 0 \times z + \mathrm{TY}$$

$$z' = 0 \times x + 0 \times y + 1 \times z + \mathrm{TZ}$$

so that the matrix describing the translation is

$$\begin{pmatrix} 1 & 0 & 0 & \mathrm{TX} \\ 0 & 1 & 0 & \mathrm{TY} \\ 0 & 0 & 1 & \mathrm{TZ} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The procedure ‘tran3’ for producing such a matrix A, given the parameters TX, TY and TZ, is given in listing 8.2.

Listing 8.2

```
9000 REM tran3
9010 DEF PROCtran3(TX,TY,TZ)
9020 A(1,1)=1 : A(1,2)=0 : A(1,3)=0 : A(1,4)=TX
9030 A(2,1)=0 : A(2,2)=1 : A(2,3)=0 : A(2,4)=TY
9040  A(3,1)=0 : A(3,2)=0 : A(3,3)=1 : A(3,4)=TZ
9050 ENDPROC
```

Scaling

The x-coordinate of every point to be transfonned is scaled by a factor SX, the y-coordinate by SY and the z-coordinate by SZ, thus

$$x' = \mathrm{SX} \times x + 0 \times y + 0 \times z + 0$$

$$y' = 0 \times x + \mathrm{SY} \times y + 0 \times z + 0$$

$$z' = 0 \times x + 0 \times y + \mathrm{SZ} \times z + 0$$

giving the matrix

$$\begin{pmatrix} \mathrm{SX} & 0 & 0 & 0 \\ 0 & \mathrm{SY} & 0 & 0 \\ 0 & 0 & \mathrm{SZ} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Usually the scaling values are positive, but if any of the va]ues are negative then this leads to a reflection as well as (possibly) scaling. For example, if SX = SZ = 1 then points are reflected in the *y*/*z* plane through the origin. A procedure 'scale3' to produce such a scaling matrix A given SX, SY and SZ is shown in listing 8.3

Listing 8.3

```
8900 REM scale3
8910 DEF PROCscale3(SX,SY,SZ)
8920 A(1,1)=SX : A(1,2)=0  : A(1,3)=0  : A(1,4)=0
8930 A(2,1)=0  : A(2,2)=SY : A(2,3)=0  : A(2,4)=0
8940 A(3,1)=0  : A(3,2)=0  : A(3,3)=SZ : A(3,4)=0
8950 ENDPROC
```

Rotation about a Coordinate Axis

In order to consider the rotation about a general axis p + mq by a given angle it is first necessary to simplify the problem by considering rotation about one of the coordinate axes.

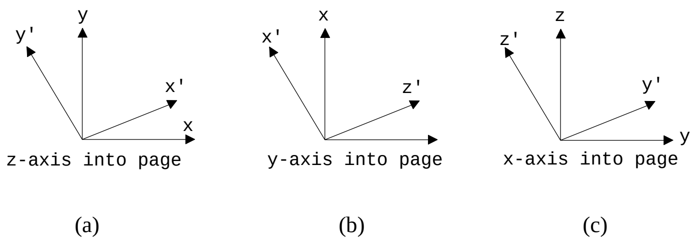

Figure 8.1

(a) Rotation by angle θ *about the x-axis*
Referring to figure 8.1c, the axis of rotation is perpendicular to the page (the positive *x*-axis being into the page), and since we are using left-handed axes the figure shows the point (*x'*, *y'*, *z'*) that results from the transformations of an arbitrary point (*x*, *y*, *z*). We see that the rotation actually reduces to a two-dimensional rotation in the *y*/*z* plane that passes through the point; that is, after the rotation the *x*-coordinate remains unchanged. By using the ideas explained in chapter 4 we get the equations

$x' = x$

$y' = \cos\theta \times y - \sin\theta \times z$

$z' = \sin\theta \times y + \cos\theta \times z$

and thus the matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin q & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(b) Rotation by an angle θ *about the y-axis*
Referring to figure 8.1b, we now have the positive *y*-axis into the page, and because of the left-handedness of the axes the positive *z*-axis is horizontal; to the right of the origin and the positive *x*-axis is above the origin. This leads us to the equations

$x' = \sin\theta \times z + \cos\theta \times x$

$y' = y$

$z' = \cos\theta \times z - \sin\theta \times x$

which gives the matrix

$$\begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(c) Rotation by an angle θ *about the z-axis*
Referring to figure 8.1a we get the equations

$x' = \cos\theta \times x - \sin\theta \times y$

$y' = \sin\theta \times x + \cos\theta \times y$

$z' = z$

and the matrix

$$\begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

A subprogram 'rot3' to produce such a mtrix A, given the angle THETA and the axis number AXIS (AXIS = 1 for the *x*-axis, AXIS = 2 for the *y*-axis and AXIS = 3 for the *z*-axis is given in listing 8.4.

Listing 8.4

```
8600 REM rot3
8610 DEF PROCrot3(THETA,AXIS)
8620 LOCAL AX1,AX2,CT,ST
8630 AX1=(AXIS MOD 3)+1
8640 AX2=(AX1 MOD 3)+1
8650 CT=COS(THETA)  : ST=SIN(THETA)
8660 A(AXIS,AXIS)=1 : A(AXIS,AX1)=0 : A(AXIS,AX2)=0
8670 A(AX1,AXIS)=0  : A(AX1,AX1)=CT : A(AX1,AX2)=-ST
8680 A(AX2,AXIS)=0  : A(AX2,AX1)=ST : A(AX2,AX2)=CT
8690 A(1,4)=0 : A(2,4)=0 : A(3,4)=0
8700 ENDPROC
```

Inverse Transformations

Before we can consider the general rotation transformation, it is necessary to look at inverse transformations. An inverse transformation returns the points transformed by a given transformation back to their original position. If a transformation is represented by a matrix A, then the inverse transformation is given by matrix A^{-1}, the inverse of A. There is no need to explicitly calculate the inverse of a matrix by using such techniques as the Adjoint method (listing 7.4): we can use listings 8.2, 8.3 and 8.4 with parameters that are derived from the parameters of the original transformation:

(1) A translation by (TX, TY, TZ) is inverted with a translation by (–TX, –TY, –TZ).
(2) A scaling by SX, SY and SZ is inverted with a scaling by 1/SX, 1/SY and 1/SZ.
(3) A rotation by an angle θ about a given axis is inverted with a rotation by an angle $-\theta$ about the same axis.
(4) If the transformation matrix is the product of a number of translation, scaling and rotation matrices $A \times B \times C \times \ldots \times L \times M \times N$, then the inverse transformation is
$$N^{-1} \times M^{-1} \times L^{-1} \times \ldots \times C^{-1} \times B^{-1} \times A^{-1}$$

Rotation of Points by an Angle γ about a General Axis $\boldsymbol{p} + \mu\boldsymbol{q}$

Assume $\boldsymbol{p} \equiv$ (PX, PY, PZ) and $\boldsymbol{q} \equiv$ (QX, QY, QZ). We break down the task into a numbe of subtasks:

(a) We translate all of space so that the axis of rotation goes through the origin. This is achieved by adding a vector $-\boldsymbol{p}$ to every point in space with a matrix F say, which is generated by a call to 'tran3' with parameters –PX, –PY and –PZ. The inverse matrix F^{-1} will be needed later and is found by a call to 'tran3' with parameters PX, PY and PZ. After this transformation the axis of rotation is the line $\boldsymbol{0} + \mu\boldsymbol{q}$ that passes through the origin.

$$F=\begin{pmatrix} 1 & 0 & 0 & -PX \\ 0 & 1 & 0 & -PY \\ 0 & 0 & 1 & -PZ \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad F^{-1}=\begin{pmatrix} 1 & 0 & 0 & PX \\ 0 & 1 & 0 & PY \\ 0 & 0 & 1 & PZ \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(b) We then rotate space about the z-axis by an angle –a, where (ALPHA =) $\alpha = \tan^{-1}$ (QY/QX), given by the matrix G. The matrix may be generated by a call to 'rot3', with parameters angle –ALPHA and axis 3, and the inverse matrix G^{-1} by a call to 'rot3' with ALPHA and 3. At this stage the axis of rotation is a line lying in the x/z plane that passes through the point (v, 0, QZ).

$$G=\frac{1}{v}\begin{pmatrix} QX & QY & 0 & 0 \\ -QY & QX & 0 & 0 \\ 0 & 0 & v & 0 \\ 0 & 0 & 0 & \mathrm{v} \end{pmatrix} \quad G^{-1}=\frac{1}{v}\begin{pmatrix} QX & -QY & 0 & 0 \\ QY & QX & 0 & 0 \\ 0 & 0 & v & 0 \\ 0 & 0 & 0 & v \end{pmatrix}$$

where v is the positive number given by $v^2 = QX^2 + QY^2$.
(c) We now rotate space about the y-axis by an angle $-\beta$, where (BETA =) $\beta = \tan^{-1}$ (v/QZ), given by the matrix H which is obtained by the call 'rot3' with parameters angle – BETA and axis 2, and the inverse matrix H–1 by a 'rot3' call with parameters BETA and 2

$$H=\frac{1}{w}\begin{pmatrix} QZ & 0 & -v & 0 \\ 0 & w & 0 & 0 \\ v & 0 & QZ & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad G-1=\frac{1}{v}\begin{pmatrix} QZ & 0 & v & 0 \\ 0 & \mathrm{w} & 0 & 0 \\ -v & 0 & QZ & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where w is the positive number given by $w^2 = v^2 + QZ^2 = QX^2 + QY^2 + QZ^2$. So the point ($v$, 0, QZ) is transformed to (0, 0, w), hence the axis of rotation is along the z–axis.
(d) We can now rotate space by an angle γ (GAMMA) about the axis of rotation by using matrix W which is generated by 'rot3' (with angle GAMMA and axis 3):

$$W = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(e) We need to return the axis of rotation to its original position so we multiply by H^{-1}, G^{-1} and finally F^{-1}.

Thus the final matrix P that rotates space by the angle γ about the axis $\boldsymbol{p} + \mu\boldsymbol{q}$ is $P = F^{-1} \times G^{-1} \times H^{-1} \times W \times H \times G \times F$. Naturally some of these matrices may reduce to the identity matrix in some special cases and can be ignored. For example if the axis of rotation goes through the origin then F and F^{-1} are identical to the identity matrix and can be ignored.

So it is possible to write a special procedure ‘genrot’ (listing 8.5) which achieves this rotation and returns the required matrix P given GAMMA, (PX, PY, PZ) and (QX, QY, QZ).

Listing 8.5

```
5000 REM genrot / rotate space about a general axis
5010 DEF PROCgenrot(PX,PY,PZ,QX,QY,QZ,GAMMA)
5020 LOCAL ALPHA,BETA
5030 PROCtran3(-PX,-PY,-PZ) : PROCmult3
5040 ALPHA=FNangle(QX,QY)
5050 PROCrot3(-ALPHA,3) : PROCmult3
5060 BETA=FNangle(QZ,SQR(QX*QX+QY*QY))
5070 PROCrot3(-BETA,2) : PROCmult3
5080 PROCrot3(GAMMA,3) : PROCmult3
5090 PROCrot3(BETA,2) : PROCmult3
5100 PROCrot3(ALPHA,3) : PROCmult3
5110 PROCtran3(PX,PY,PZ) : PROCmult3
5120 ENDPROC
```

Example 8.1

What happens to the points (0, 0, 0), (1 , 0, 0), (0, 1, 0), (0, 0, 1) and (1, 1, 1) if space is rotated by $\pi/4$ radians about an axis $(1, 0, 1) + \mu(3, 4, 5)$.

Using the above theory we note that

$$F = \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad F^{-1} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$G = \frac{1}{5}\begin{pmatrix} 3 & 4 & 0 & 0 \\ -4 & 3 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{pmatrix} \quad G = \frac{1}{5}\begin{pmatrix} 3 & -4 & 0 & 0 \\ 4 & 3 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{pmatrix}$$

$$H = \frac{1}{\surd 2}\begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & \surd 2 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & \surd 2 \end{pmatrix} \quad H^{-1} = \frac{1}{\surd 2}\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & \surd 2 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \surd 2 \end{pmatrix}$$

$$W = \frac{1}{\surd 2}\begin{pmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & \surd 2 & 0 \\ 0 & 0 & 0 & \surd 2 \end{pmatrix} \text{ and}$$

$$P = \frac{1}{50\surd 2}\begin{pmatrix} 41 + 9\surd 2 & -12 - 13\surd 2 & -15 + 35\surd 2 & -26 + 6\surd 2 \\ -12 + 37\surd 2 & 34 + 16\surd 2 & -20 + 5\surd 2 & -26 + 6\surd 2 \\ -15 - 5\surd 2 & -20 + 35\surd 2 & 25 + 25\surd 2 & -10 + 30\surd 2 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

where $P = F^{-1} \times G^{-1} \times H^{-1} \times W \times H \times G \times F$ is the matrix representation of the required transformation. Premultiplying the column vectors equivalent to (0, 0, 0), (1, 0, 0), (0, 1, 0), (0, 0, 1) and (1, 1, 1) by P and changing the resulting column vectors back into row form and taking out a factor $1/50\surd 2$ gives the coordinates $(-26 + 6\surd 2, 32 - 42\surd 2, -10 + 30\surd 2)$, $(15 + 15\surd 2, 20 - 5\surd 2, -25 + 25\surd 2)$, $(-38 - 7\surd 2, 66 - 26\surd 2, -30 + 65\surd 2)$, $(-41 + 41\surd 2, 12 - 37\surd 2\ 15 + 55\surd 2)$ and $(-12 + 37\surd 2, 34 + 16\surd 2, -20 + 85\surd 2)$ respectively. Naturally, translating and rotating space should leave relative positions unchanged; in particular the angles between direction vectors should be unchanged (the same cannot be said about the scaling transformation which in general does alter relative positions). In the original system the three lines from (0, 0, 0) to (1, 0, 0), (0, 1 , 0) and (0, 0, 1), respectively, are mutually perpendicular (that is, the dot product of pairs of these directions should be zero). The dot product of the directions in the transformed system should also be zero: the three directional vectors (with $1/50\surd 2$ vectored out) are $(41 + 9\surd 2, -12 + 37\surd 2, -15 - 5\surd 2)$, $(-12 - 13\surd 2, 34 + 16\surd 2, -20 + 35\surd 2)$ and $(-15 + 35\surd 2, -20 + 5\surd 2, 25 + 25\surd 2)$, and the dot product of any pair is zero.

Similarly the dot product of the direction vector from the origin to (1, 1, 1) in the original system, taken with any of the original directions above, gives the same value (= 1). This is also tme in the transformed system: the fourth direction is $(14 + 31\sqrt{2}, 2 + 58\sqrt{2}, -10 + 55\sqrt{2})$, and when we take the dot product with each of the three direction vectors above we get the value 5000, which when we take into account the factor $(1/50\sqrt{2})^2$ gives the value 1.

A program that reads in the axis ofrotation (PX, PY, PZ) + μ(QX, QY, QZ) and the angle GAMMA, and rotates any point (XX, YY, ZZ) about this axis by an angle GAMMA is given in listing 8.6.

Listing 8.6

```
100 REM Rotation about given axis
110 DIM A(4,4),B(4,4),R(4,4)
119 REM read in data on rotation
120 CLS : PRINT TAB(0,3),"Rotation about given axis",SPC(10)
130 INPUT"Base vector of axis ",PX,PY,PZ
140 INPUT"Direction vector of axis ",QX,QY,QZ
150 INPUT"Angle of rotation ",GAMMA
160 CLS
170 PRINT TAB(0,3);"Base vector of axis "
180 PRINT TAB(0,4);"(";PX;",";PY;",";PZ;")"
190 PRINT TAB(0,6);"Direction vector of axis "
200 PRINT TAB(0,7);"(";QX;",";QY;",";QZ;")"
210 PRINT TAB(0,9);"Angle of rotation "
220 PRINT TAB(0,10);GAMMA
229 REM calculate rotation matrix R
230 PROCidR3 : PROCgenrot(PX,PY,PZ,QX,QY,QZ,GAMMA)
239 REM input point (XX,YY,ZZ)
240 FOR I%=13 TO 21 : PRINT TAB(0,I%);SPC(40) : NEXT I%
250 PRINT TAB(0,12);"Coordinates of point"
260 INPUT XX,YY,ZZ
270 PRINT TAB(0,13);"(";XX;",";YY;",";ZZ;")"
279 REM (XX,YY,ZZ) becomes (RX,RY,RZ)
280 RX=R(1,1)*XX+R(1,2)*YY+R(1,3)*ZZ+R(1,4)
290 RY=R(2,1)*XX+R(2,2)*YY+R(2,3)*ZZ+R(2,4)
300 RZ=R(3,1)*XX+R(3,2)*YY+R(3,3)*ZZ+R(3,4)
310 PRINT TAB(0,15);"become"
320 PRINT TAB(0,17);"(";RX;",";RY;",";RZ;")"
330 PRINT TAB(0,21);"press any key to continue"
340 IF NOT INKEY(0) THEN PRINT TAB(0,20);SPC(40) : GOTO 240 ELSE
340
```

Exercise 8.1

Experiment with these ideas. You can always make a check on your final transformation matrix by considering simple values as above, and you can use the previous listings to check your answer. It is essential that you are confident in the use of matrices, and the best way to get this confidence is to experiment. You will make lots of arithmetic errors initially, but you will soon come to think of transformations in terms of their matrix representation, and this will greatly ease the study of drawing three-dimensional objects.

Exercise 8.2
You will have noticed that the procedure ‘rot3’ is usually called with THETA generated by ‘angle’ which uses values AX and AY as input parameters. ‘rot3’ calculates the cosine and sine of angle THETA - but we know these are $AX/\sqrt{(AX^2 + AY^2)}$ and $AY/\sqrt{(AX^2 + AY^2)}$ respectively. Write another rotation procedure ‘rotxy’ that calculates the rotation matrix direction from AX and AY without resorting to ‘angle' .

Exercise 8.3
In chapter 4 we noted that some writers use row rather than column vectors, and postmultiply rather than premultiply. We decided against this interpretation so that the matrix of a transformation would correspond directly with the coefficients of the transformation equations. In this other interpretation it is the transpose of the matrix that is identical to the coefficients. It is useful to be aware of this other method, so use it to rewrite all the programs given in this chapter (and the remainder of this book). Remember though, it is not important which method you finally decide to use *as long as you are consistent*. We have used the column vector notation because we have found it causes less confusion in the early stages oflearning the subject!

Complete Programs

I All the listings in this chapter, 8.1 (' mult3’ and ‘idR3'), 8.2 (' tran3'), 8.3 (' scale3'), 8.4 (' rot3'), 8.5 (' genrot'), 8.6 (' main program') and listing 3.3 (' angle'). Required data: base vector (PX, PY, PZ) and direction vector (QX, QY, QZ) of the axis of rotation and the angle GAMMA. Then any number of three-dimensional coordinates (XX, YY, ZZ). Try (0, 0, 0), (1 ,1, 1) and $\pi/4$, and points (1, 0,1), (1 ,1, 1), (1, 2, 3).