Hardware systems which use true floating-point representations have less need of homogeneous coordinates and for these N and W may normally be set to unity.

3.3.1.1.3. *Notation*

In this chapter the conventions of matrix algebra will be adhered to except where it is convenient to show operations on elements of vectors, matrices and tensors, where a subscript notation will be used with a modified summation convention in which summation is over lower-case subscripts *only*. Thus the equation

$$x_I = A_{Ij}X_j$$

is to be read 'For any I, x_I is $A_{Ij}X_j$ summed over j'.

Subscripts using the letter i or later in the alphabet will relate to the usual three dimensions and imply a three-term summation. Subscripts a to h are not necessarily so limited, and, in particular, the subscript a is used to imply summation over atoms of which there may be an arbitrary number.

We shall use the superscript T to denote a transpose, and also use the Kronecker delta, δ_{IJ}, which is 1 if $I = J$ and zero otherwise, and the tensor ε_{IJK} which is 1 if I, J and K are a cyclic permutation of 1, 2, 3, -1 if an anticyclic permutation, and zero otherwise.

$$\varepsilon_{IJK} = (I-J)(J-K)(K-I)/2 \quad 1 \le I, J, K \le 3.$$

A useful identity is then

$$\varepsilon_{iJK}\varepsilon_{iLM} = \delta_{JL}\delta_{KM} - \delta_{JM}\delta_{KL}.$$

Single modulus signs surrounding the symbol for a square matrix denote its determinant, and around a vector denote its length.

The symbol $\simeq$ is defined in the previous section.

3.3.1.1.4. *Standards*

The sections of this chapter concerned with graphics are primarily concerned with the mathematical aspects of graphics programming as they confront the applications programmer. The implementations outlined in the final section have all, so far as the author is aware, been developed *ab initio* by their inventors to deal with these aspects using their own and unrelated techniques and protocols. It is clear, however, that standards are now emerging, and it is to be hoped that future developments in applications software will handle the graphics aspects through one or other of these standards.

First among these standards is the Graphical Kernel System, GKS, defined in *American National Standards Institute, American National Standard for Information Processing Systems – Computer Graphics – Graphical Kernel System (GKS) Functional Description* (1985) and described and illustrated by Hopgood *et al.* (1986) and Enderle *et al.* (1984). GKS became a full International Standards Organization (ISO) standard in 1985, and its purpose is to standardize the interface between application software and the graphics system, thus enhancing portability of software. Specifications for Fortran, Pascal and Ada formulations are at an advanced stage of development. Its value to crystallographers is limited by the fact that it is only two-dimensional. A three-dimensional extension known as GKS-3D, defined in *International Standards Organisation, International Standard Information Processing Systems – Computer Graphics – Graphical Kernel System for Three Dimensions (GKS-3D), Functional Description* (1988) became an ISO standard in 1988. Perhaps of greatest interest to crystallographers, however, is the Programmers' Hierarchical Interactive Graphics System (PHIGS) (Brown, 1985; Abi-Ezzi & Bunshaft, 1986) since this allows hierarchical segmentation of picture content to exist in both the applications software and the graphics device in a related manner, which GKS does not. Some graphics devices now available support this type of working and its exploitation indicates the choice of PHIGS. Furthermore, Fortran implementations of GKS and GKS-3D require points to be stored in arrays dimensioned as $X(N)$, $Y(N)$, $Z(N)$ which may be equivalenced (in the Fortran sense) to $XYZ(N, 3)$ but not to $XYZ(3, N)$, which may not be convenient. PHIGS also became an International Standard in 1988: *American National Standards Institute, American National Standard for Information Processing Systems – Computer Graphics – Programmer's Hierarchical Graphics System (PHIGS) Functional Description, Archive File Format, Clear-Text Encoding of Archive File* (1988). PHIGS has also been extended to support the capability of raster-graphics machines to represent reflections, shadows, see-through effects *etc.* in a version known as PHIGS+ (van Dam, 1988).

Increasingly, manufacturers of graphics equipment are orienting their products towards one or other of these standards. While these standards are not the subject of this chapter it is recommended that they be studied before investing in equipment.

In addition to these standards, related standards are evolving under the auspices of the ISO for defining images in a file-storage, or metafile, form, and for the interface between the device-independent and device-dependent parts of a graphics package. Arnold & Bono (1988) describe the ANSI and ISO Computer Graphics Metafile standard which provides for the definition of (two-dimensional) images. The definition of three-dimensional scenes requires the use of (PHIGS) archive files.

3.3.1.2. *Orthogonal (or rotation) matrices*

It is a basic requirement for any graphics or molecular-modelling system to be able to control and manipulate the orientation of the structures involved and this is achieved using orthogonal matrices which are the subject of these sections.

3.3.1.2.1. *General form*

If a vector $\mathbf{v}$ is expressed in terms of its components resolved onto an axial set of vectors $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ which are of unit length and mutually perpendicular and right handed in the sense that $(\mathbf{X} \times \mathbf{Y}) \cdot \mathbf{Z} = +1$, and if these components are v_I, and if a second set of axes $\mathbf{X}'$, $\mathbf{Y}'$, $\mathbf{Z}'$ is similarly established, with the same origin and chirality, and if $\mathbf{v}$ has components v'_I on these axes then

$$v'_I = a_{Ij}v_j,$$

in which a_{IJ} is the cosine of the angle between the Ith primed axis and the Jth unprimed axis. Evidently the elements a_{IJ} comprise a matrix $\boldsymbol{R}$, such that any row represents one of the primed axial vectors, such as $\mathbf{X}'$, expressed as components on the unprimed axes, and each column represents one of the unprimed axial vectors expressed as components on the primed axes. It follows that $\boldsymbol{R}^T = \boldsymbol{R}^{-1}$ since elements of the product $\boldsymbol{R}^T\boldsymbol{R}$ are scalar products among perpendicular unit vectors.

A real matrix whose transpose equals its inverse is said to be *orthogonal.*

Since $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ can simultaneously be superimposed on $\mathbf{X}'$, $\mathbf{Y}'$ and $\mathbf{Z}'$ without deformation or change of scale the relationship is one of rotation, and orthogonal matrices are often referred to as rotation matrices. The operation of replacing the vector $\mathbf{v}$ by $\boldsymbol{R}\mathbf{v}$ corresponds to rotating the axes from the unprimed to the primed set with $\mathbf{v}$ itself unchanged. Equally, the same operation corresponds to retaining fixed axes and rotating the vector in the opposite sense. The second interpretation is the one more frequently helpful since conceptually it corresponds more closely to rotational operations on objects, and it is primarily in this sense that the following is written.

If three vectors $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$ form the edges of a parallelepiped, then its volume V is

Copyright © 2006 International Union of Crystallography