The package `matrix.inc`

Unless noted otherwise, all matrices are single arrays, laid out in columns. Vectors are row vectors, matrices usually apply on the right, to conform with PostScript conventions. Thus transformations can be applied in RPN, in succession:

```
v A vector-matrix B vector-matrix
```

Most routines work in any number of dimensions.

• `to-single-array`

Input: a double array
Output: all its elements laid out in a single array

• `to-double-array`

Input: a single array of size n^2
Output: square matrix in a single array

• `identity`

Input: an integer $n \geq 0$
Output: the $n \times n$ identity matrix

• `dot-product`

Input: two vectors u and v
Output: their dot product

The dimension of v must be at least as large as that of u. If u has length n it will return $\sum_0^{n-1} u_i v_i$.

• `vector-scale`

Input: v and c
Output: cv

• `vector-add`

Input: u and v
Output: $u + v$

Sums over u.

• `vector-sub`

Input: u and v
Output: $u - v$

Sums over u.

• `vector-length`

Input: v
Output: $\|v\|$

Tries to be avoid overflow, first dividing by entry of maximum size.

• `normalized`

Input: v
Output: $v/\|v\|$

Calls `vector-length`.

- `normalized3d`

Input: v
Output: $v/\|v_3\|$ where v_3 is the vector of the first three coordinates. Used for lighting in 3D routines.

- `cross-product`

Input: u and v
Output: $u \times v$

- `rotation-matrix3d`

Input: axis α and angle A
Output: the matrix of rotation.

The axis does not have to be normalized.

The axis α does not have to be normalized.

- `reflection-matrix3d`

Input: (A, B, C)
Output: Matrix of reflection perpendicular to input (the normal).

The normal does not have to be normalized.

- `euclidean-reflect`

Input: a and v
Output: $v - 2(a \bullet v)/\|a\|^2$

- `matrix-mul`

Input: a and b Out: ab

Both a and b must be square matrices.

- `matrix-vector`

Input: a matrix $A = [\,\chi_1\ \ldots\ \chi_c]$ and vector v
Output: Av

- `vector-matrix`

Input: v and a matrix $A = [\,\chi_1\ \ldots\ \chi_c]$
Output: vA

In this and the previous routine, dimensions must be match and be positive.

- `transpose`

Input: a square matrix A
Output: transpose of A

- `3x3-det`

Input: array of length 9
Output: determinant computed by explicit formula

- `3x3-inverse`

Input: array of length 9
Output: inverse matrix computed by explicit formula

- `acos`

Input: $-1 \le t \le 1$
Output: $\arccos(t)$ in degrees

- `angle-between`

Input: u and v
Output: angle in degrees

- `skew-reflect`

Input: x, α, and $\alpha^\vee$
Output: $x - \langle \alpha, x \rangle \, \alpha^\vee$

- `skew-reflection-matrix`

Input: α, and $\alpha^\vee$
Output: the matrix of the reflection