MATRIXUSE

Interlisp-D contains low-level functions for high-speed matrix-multiplication. The MatrixUse LispLibrary package provides functions that let you easily manipulate the contents of these matrices.

(CREATE1BY3) [Function]

Returns an ARRAY that appears to the matrix multiplication functions to be a 1 by 3 vector. The phrase "appears ... to be a 1 by 3 vector" is due to the fact that Interlisp-D arrays are currently one dimensional; the MatrixUse package transparently folds two dimensions into one.

(CREATE1BY4) [Function]

Returns a 1 by 4 vector.

(CREATE3BY1) [Function]

Returns a 3 by 1 vector.

(CREATE3BY3) [Function]

Returns a 3 by 3 matrix.

(CREATE4BY1) [Function]

Returns a 4 by 1 vector.

(CREATE4BY4) [Function]

Returns a 4 by 4 matrix.

(IDENTITY3BY3 *M*) [Function]

Returns a 3 by 3 identity matrix (a matrix with the elements of its main diagonal all set to unity). In this and all following MatrixUse functions, if *M* is supplied, sets it to be the identity matrix and returns it.

(IDENTITY4BY4 *M*) [Function]

Returns a 4 by 4 identity matrix.

(ROTATE3BY3 *THETA RADIANSFLG M*) [Function]

Returns a 3 by 3 matrix that, when multiplied by a vector, will rotate that vector an angle of *THETA* around the origin. *THETA* is measured in degrees or radians, depending on *RADIANSFLG*.

(ROTATE4BY4.ABOUTX *THETA RADIANSFLG M*) [Function]

Returns a 4 by 4 matrix suitable for rotating a point in 3-space around the X-axis by *THETA.*

(ROTATE4BY4.ABOUTY *THETA RADIANSFLG M*) [Function]

Returns a 4 by 4 matrix suitable for rotating a point in 3-space around the Y-axis by *THETA.*

(ROTATE4BY4.ABOUTZ *THETA RADIANSFLG M*) [Function]

Returns a 4 by 4 matrix suitable for rotating a point in 3-space around the Z-axis by *THETA.*

(SCALE3BY3 *Sx Sy M*) [Function]

Returns a 3 by 3 matrix that will scale by a factor of *Sx* in the X-axis and *Sy* in the Y-axis*.*

(SCALE4BY4 *Sx Sy Sz M*) [Function]

Returns a 4 by 4 matrix that will scale by a factor of *Sx* in the X-axis, *Sy* in the Y-axis, and *Sz* in the Z-axis*.*

(TRANSLATE3BY3 *Tx Ty M*) [Function]

Returns a 3 by 3 matrix that will translate by a distance of *Tx* in the X-axis and *Ty* in the Y-axis.

(TRANSLATE4BY4 *Tx Ty Tz M*) [Function]

Returns a 4 by 4 matrix that will translate by a distance of *Tx* in the X-axis, *Ty* in the Y-axis, and *Tz* in the Z-axis*.*

(SET.XCOORD *VECTOR VALUE*) [Function]

Given a vector *VECTOR* of length 3 or 4, sets the X component to be *VALUE*.

(SET.YCOORD *VECTOR VALUE*) [Function]

Sets the Y component of *VECTOR* to be *VALUE*.

(SET.ZCOORD *VECTOR VALUE*) [Function]

Sets the Z component of *VECTOR* to be *VALUE*.

(SET.WCOORD *VECTOR VALUE*) [Function]

Sets the W (fourth) component of *VECTOR* to be *VALUE*.

(GET.XCOORD *VECTOR*) [Function]

Returns the X component of *VECTOR.*

(GET.YCOORD *VECTOR*) [Function]

Returns the Y component of *VECTOR.*

(GET.ZCOORD *VECTOR*) [Function]

Returns the Z component of *VECTOR.*

(GET.WCOORD *VECTOR*) [Function]

Returns the W component of *VECTOR.*