LinCalc v1.0 for Windows 95 and Windows NT 4.0

Help Contents

Overview

LinCalc is an object-oriented calculating system designed to make working with vectors and matrices as simple as possible. Each vector or matrix that is entered, and the result of every operation, becomes an "object" displayed in its own persistent window. The user performs an operation on one or more objects by selecting the object windows and then activating the appropriate function button on the calculator face. Calculations can easily be chained, because the result of an operation becomes a new object. Objects can be entered by the user or loaded from files, and they can also be stored into files. The system has 16 vector and matrix functions, and can operate with objects that have row and column dimensions of up to 20 elements.

Working With Objects

Creating New Objects

To Create a new object, first select the dimensions of the object using the two drop down list boxes, and then activate either the Zeroed Object or the Random Object buttons. The Zeroed Object button creates an object that has its elements initialized to zero, and displays the object with empty elements. The Random Object button creates and displays an object with its elements initialized to random numbers between zero and one. Once an object has been created, its elements can be edited. Click on an element in an object to move the cursor to it, or use the tab key to navigate between elements in an object. Elements that are left blank will be treated as zeroes by LinCalc.

Selecting Objects

It is necessary to select an object before an operation can be performed on it. To select an object, right-click on the background of its window. The window background will change color to indicate selection. To deselect an object, right click again on its background. Since no operation requires more than two arguments, no more than two objects can be selected at once. The order of selection determines the order in which an operation is performed - the first selected object is the left operand. Selection is removed from argument objects once the operation is performed.

Working With Object Windows

Objects windows can be moved anywhere on the screen or minimized, maximized, or destroyed. To destroy all existing objects at once, activate the Close All Windows button. The caption of an object window contains its object number and the dimensions of the object. If an object was loaded from a file, its caption includes the file name. If an object was created by a function, its caption includes the function and its argument(s). If these argument objects were themselves generated by functions, then those functions are also given with their arguments. In this way, the caption of window specifies the chain of operations that created it. LinCalc keeps track of all of the existing object windows by displaying their numbers in the large list box on the left side of the calculator face. Select one or more object windows in this list to bring windows that have become hidden back to the front.

Vector Functions

Magnitude (Mag)
Dot Product (Dot)
Cross Product (Crs)
Angle (Ang)

Matrix Functions

Add (Add)
Subtract (Sub)
Multiply (Mul)
Power (Pow)
Transpose (Trp)
Cofactors (Cof)
Adjoint (Adj)
Inverse (Inv)
Determinant (Det)
Trace (Trc)
Compare (Cmp)
Solve (Slv)

Scalars

All of the matrix functions accept [1x1] objects as legal arguments that represent scalars. This means that **Add**, **Sub**, **Mul**, **Pow**, and **Inv** can be used with scalar arguments. The **Inv** function applied to a scalar returns the multiplicative inverse, so a matrix can be divided by a scalar by multiplying it by the inverse of the scalar. The vector functions **Mag**, **Dot**, and **Ang** treat a [1x1] object as a one dimensional vector.

File Operations

To save an object to a file, select it and activate the File Save function. To load an object from a file, activate the File Load Function. The file that an object is loaded from must have the following format:

The first line must contain two integers: the number of rows in the object and the number of columns in the object. These integers must have values between 1 and 20. The line must be terminated by a “|” character. Subsequent lines must contain the element values for each row of the object. The values must be seperated by spaces. Each row must be terminated by a “|” character. It is not required that this file be organized into lines terminated by newline characters, but if no newline characters are present, at least one space must be inserted after each “|” character.

Performance And Limitations

LinCalc performs all calculations in double precision, but displays ten significant figures in results. Chained calculations are effected by this precision limitation. Partial pivoting is used to reduce roundoff error in the determinant routine, and in the portion of the inverse routine in which the matrix is reduced to row-echelon form. An error message is given if a pivot is as small as 1E-8, but the system is still subject to the large errors that can occur if, for example, the determinant is small. The accuracy of a calculation can easily be checked, however, by performing further calculations. To check an inverse, for example, just select the inverse along with the original matrix, multiply the two, and compare the result with the identity matrix. To the check the solution of a linear system, just select the solution vector along with the coefficient matrix, multiply the two, select this result along with the original vector of constants, subtract the two, and examine the final result.

The largest object allowed in LinCalc has 20 rows and 20 columns. The total number of elements in all objects is limited to 5000 to avoid hitting system limits. This allows up to 50 10x10 objects, for example. When this limit is reached, a warning message will be displayed, and the user may close some objects to free space.

The performance of LinCalc will, of course, depend upon the hardware that it is run on. Gaussian elimination is used to calculate the determinant, and Gauss-Jordan elimination is used to calculate the inverse. On a 200 MHZ Pentium Pro machine running Windows NT 4.0, a
20x20 matrix inverted in about 1 second.

Magnitude (Mag)

This function takes either a [1xn] or a [nx1] object as an argument, and returns the magnitude of the vector.

Dot Product (Dot)

This function takes two [1xn] or [nx1] objects as arguments and returns the dot product of the two vectors.

Cross Product (Crs)

This function takes two [1x3] or [3x1] objects as arguments and returns the the cross product of the two vectors with the same dimensions as the arguments. It is better to use row vectors if possible in this operation, because the caption will be mire visible.

Angle (Ang)

This function takes two [1xn] or [nx1] objects as arguments and returns the angle between the two vectors in radians.

Add (Add)

This function takes two [mxn] objects as arguments and returns the [mxn] sum of the two matrices.

Subtract (Sub)

This function takes two [mxn] objects as arguments and returns the [mxn] difference of the two matrices.

Multiply (Mul)

This function takes a [mxn] object and a [nxp] object as arguments and returns their [mxp] matrix product. Vectors are legal as operands. The function also accepts a [mxn] object and a [1x1] object for scalar multiplication.

Power (Pow)

This function takes a [nxn] object and a [1x1] object as arguments and returns the matrix argument raised to the power of the scalar argument.

Transpose (Trp)

This function takes a [mxn] object as its argument and returns the [nxm] transpose of the matrix.

Cofactors (Cof)

This function takes a [nxn] object as its argument and returns the [nxn] cofactors of the matrix.

Adjoint (Adj)

This function takes a [nxn] object as its argument and returns the [nxn] adjoint of the matrix.

Inverse (Inv)

This function takes a [nxn] object as its argument and returns the [nxn] inverse of the matrix. The operation returns an error message during row reduction if a pivot entry is as low as 1E-8. This is a very liberal restriction, and does not prevent large errors from occuring in the inverse if, for example, the determinant is small. To check the accuracy of an inverse, just select the inverse along with the original matrix, multiply the two, and compare the result with the identity matrix.

Determinant (Det)

This function takes a [mxn] object as its argument and returns the [1x1] determinant of the matrix. If, during row reduction, a pivot entry is as low as 1E-8, the determinant function immediately returns a zero determinant.

Trace (Trc)

This function takes a [mxn] object as its argument and returns the [1x1] trace of the matrix.

Compare (Cmp)

This function takes two [mxn] objects as its arguments and notifies the user as to whether the two arguments are identical.

Solve (Slv)

This function takes a [mxn] object and a [nx1] object as arguments and returns the solution to the linear system for which the first argument is the matrix of coefficients and the second argument is the vector of constants. The operation returns an error message during row reduction if a pivot entry is as low as 1E-8, but this does not prevent the large errors that can occur if, for example, the determinant is small. To the check the solution of a linear system, just select the solution vector along with the coefficient matrix, multiply the two, select this result along with the original vector of constants, subtract the two, and examine the final result.