Utility Functions

Hlookup(z, A, r, modifier) Takes a real input z, an array **A** of mixed real, complex, or string values, and a comparison criterion modifier. **Hlookup** matches z in the first row of **A**, subject to the conditions of modifier, and returns the result(s) in row **r** in the same columns as the matched elements.

z must be a scalar, unless you specify "range" as the comparison criterion, in which case it can be a 2-element column vector.

Lookup(z, A, B, modifier) Takes a real input z, two arrays **A** and **B** of mixed real, complex, or string values, and a comparison criterion modifier. **Lookup** matches z in **A**, subject to the conditions of modifier, and returns the value(s) in the same position(s) (that is, with the same row and column numbers) in matrix **B**.

z must be a scalar, unless you specify "range" as the comparison criterion, in which case it can be a 2-element column vector.

Match(z, A, modifier) Takes a vector or matrix **A** of real, complex, or string values, a real input z to search for in **A,** and a comparison criterion modifier, and returns a vector of indices for all matching elements in **A**.

z must be a scalar, unless you specify "range" as the comparison criterion, in which case it can be a 2-element column vector.

VHlookup(z1, z2, A, modifier) Takes two values z1 and z2, an array **A** of real, complex, or string values, and a comparison criterion modifier. **VHlookup** matches z1 in the first column of **A** and z2 in the first row of **A**, and returns the value at the intersection, subject to the conditions of modifier.

With **VHlookup**, either z1 or z2 can be a string, but not both. If either is a string, an exact match must be made in the corresponding row/column of **A**.

vhlookup(z1, z2, A) Takes two values, z1 and z2, and an array **A** of real, complex, or string values. **vhlookup** matches z1 in the first column of **A** and z2 in the first row of **A**, and returns the value at the intersection. The accuracy of the match is controlled by **TOL**.

Vlookup(z, A, c, modifier) Takes a real input z, an array **A** of mixed real, complex, or string values, and a comparison criterion modifier. **Vlookup** matches z in the first column of **A**, subject to the conditions of modifier, and returns the result(s) in column **c** in the same rows as the matched elements.

z must be a scalar, unless you specify "range" as the comparison criterion, in which case it can be a 2-element column vector.

Comparison Modifiers For Match And Lookup Functions

The following modifiers are supported by the **Match** and **lookup** functions provided as part of the Data Analysis Extension Pack:

Comparison	Meaning
"near"	Returns the value closest to z.
"gt"	Matches everything greater than the value z.
"lt"	Matches everything less than the value z.

"geq"	Matches everything greater than or equal to z.
"leq"	Matches everything less than or equal to z.
"not"	Matches everything not equal to z.
"range"	Matches everything in the specified range. z must be a two element vector containing the upper and lower bounds of the range.

localmax(M, [w]) Takes a real n x 2 matrix **M** and optional integer w, = 1, and finds the local maxima in the second column of M. w (window width) is optional, and defaults to 1; for $w > 1$, each point in the data must be greater than the surrounding (2w + 1) points to be considered a local maximum to avoid false positives from noise. Function returns a 2-column matrix of the x and y values for each maxima.

localmax(x, y, M, [w]) The 3D version of this function takes vectors of x and y coordinates, a matrix of associated z values, **M**, and performs the same search for local maxima by nearest-neighbor comparison. **M** must have the same number of rows as x, and the same number of columns as y has rows.

localmin(M, [w]) Takes a real n×2 matrix **M** and optional integer w, = 1, and finds the local minima in the second column of **M**. w (window width) is optional, and defaults to 1; for $w > 1$, each point in the data must be less than the surrounding (2w + 1) points to be considered a local minimum. Function returns a 2-column matrix of the x and y values for each minima.

localmin(x, y, M, [w]) The 3D version of this function takes vectors of x and y coordinates, a matrix of associated z values, **M**, and performs the same search for local minima by nearest-neighbor comparison. **M** must have the same number of rows as x, and the same number of columns as y has rows.