LEASQ

LEASQ returns a least squares approximated value in two or more dimensions.

Syntax

1. LEASQ(Pno%, Ndim%, "ColLab$_1",.., "ColLab$_Ndim%", Xint_1,..,Xint_Ndim%-1, Order%_1,..,Order%_Ndim%-1, [Xtrap%=0,1])
2. LEASQ(0, Npoints%, x_1, y_1, x_2, y_2,.., x_n, y_n, xint, [Xtrap%=0,1])
3. LEASQ(@ObjFn(..), Ndim%, @ObjColPar_1,.., @ObjColPar_Ndim%, Xint_1,..,Xint_Ndim%-1, Order%_1,..,Order%_Ndim%-1, [Xtrap%=0,1])
4. LEASQ(Telitab$, Ndim%, "ColLab$_1",.., "ColLab$_Ndim%", Xint_1,..,Xint_Ndim%-1, Order%_1,..,Order%_Ndim%-1, [Xtrap%=0,1]

Arguments

- Pno% is the number that refers to the TeLiTab sets in the Data slot. Pno% should be an integer value or a parameter which is assigned an integer value and is the number of the TeLiTab set in the expressions' data slot.
- Npoints% is the number of points (x,y) that are given in direct definition.
- @ObjFn() refers to the Object from which data will be used.
- TeLiTab$ refers to the string parameter that contains the TeLiTab.
- Ndim% is the number of dimensions.
- "ColLab$_1" and @ObjColPar_1 refer to the column that will be used as the parameter X in the interpolation.
- "ColLab$_2" and @ObjColPar_2 refer to the column that will be used as the parameter Y in the interpolation. etc.
- Order% is the parameter for interpolation (see above).
- Xtrap% is an optional argument to switch off a warning in the event of extrapolation. If Xtrap% = 1 no warning is given. The value is not required and is 0 by default (not suppressing the warning).

Remarks

1. See also Telitab access for a generic description on the use of TeLiTab data
2. Similar to other Data analysis functions, the LEASQ is a convenient way to evaluate data. Please also look at these functions for syntax examples
3. The LEASQ function provides a convenient way to manipulate curve data. The routine contains a least squares approximator which fits a polynomial through a series of (x,y[z,,,]) in such way that the root mean squares of the difference between the actual points and the approximation is minimal. The polynomials are defined with their order for each dimension above one, so for a 2D (x,y) curve, one order is provided. The order for each dimension should be smaller than the number of ordinates-1 in that dimension.
4. Ndim% and Order% are related to eachother. This can be simply understood by the following 2D example:

- a line can be drawn through 2 points, Order%=1 will provide an exact fit, Order%=0 will give a horizontal line between the two points which is a least squares approximation
- a parabola can be drawn through 2 points not positioned on one straight line, Order%=2 will therefore provide an exact fit, Order%=1 will give a regression line through the 'cloud' of three points which is a least squares approximation
- the same is valid for more points, higher order curves can then be used for a least squares approximation.

1. In contradistinction with the 'real' interpolators like DQUAD, SPLINT, LININT and DISINT, the LEASQ function allows points with the same X value and the points need not to be sorted in the 2D case. For multi-dimensional approximation a sorted matrix is required. See the example at the bottom of this paragraph in which power is interpolated in a matrix of power and propeller rotation rate on the basis of ship speed and propeller pitch setting.
2. The function returns the approximated value according to the selected fitting order. Extrapolation outside the x-range is performed according to the selected order of the curve for that dimension. The theoretical maximum order can be calculated using the following example relation:

   Least_square_order=ROUND(SQRT(ORCA(TeLiTab_to_be_used#)),0)
3. Please note that by extrapolating you will always run the risk of unreliable results, even when you have detemined an order for a fit based on the amount of points (ORCA(TeLiTab_to_be_used#). Always try to determine a realistic order on forehand, for instance based on physics.

Examples

Syntax 1: TeLiTab in Dataslot

In this example syntax 1 is used, the TeLiTab containing the data is addressed in the Dataslot of the relation.
The syntax is

LEASQ(Pno%, Ndim%, "ColLab$_1" or ColNo%_1,.., "ColLab$_Ndim%" or ColNo%_Ndim%, Xint_1,..,Xint_Ndim%-1, Order%_1,..,Order%_Ndim%-1, [Xtrap%=0,1])

For two dimensional interpolation on a regression line (Order%=1) and a TeLiTab in the Dataslot, the argument sequence is the following:

y = LEASQ(1, 2, "XC", "YC", x, 1, 1)

With the following Telitab set in the Data slot:

```
|LEASQ1|
0
2 "XC" "YC"
"1" 1 1
"2" 2 4
"3" 6 9
"4" 8 16
"5" 12 25
"6" 12 36
"7" 16 49
"8" 17 64
"9" 17 81
"10" 10 100|
```

This relation returns for x = 12 a value of

y=46.45

Remarks

1. In case you apply the symbolic addressing of the columns for the description of the point on the curve or surface to compute the differential for, e.g. "Par_x" and "Par_y", please make sure that your Telitab set contains these names. If not, an error message is generated and the calculation is stopped.
2. If you use |1| as data header instead of |LEASQ1| it means that any special function in that expression referring to a data set Pno%=1 can access that data set.
3. In addition to traditional two-dimensional interpolation it is also possible to perform matrix (n-dimensional) interpolation. In that case, a special Telitab set definition format can be applied, clarified by the following database which is an example of an interpolation in a matrix of power and propeller rotation rate on the basis of ship speed and propeller pitch setting (for the sake of compactness, the data set size was reduced):

```
| 0
4 "VS" "PDRA" "N" "PS"
17.0 1.0 157.5 2300
18.0 1.0 170.0 2970
19.0 1.0 184.5 3940
20.0 1.0 199.0 5140
17.0 1.1 147.0 2255
18.0 1.1 159.0 2930
19.0 1.1 173.0 3900
20.0 1.1 187.5 5110
17.0 1.2 138.5 2245
18.0 1.2 150.3 2925
19.0 1.2 163.5 3915
20.0 1.2 177.5 5150
17.0 1.3 131.5 2250
18.0 1.3 142.5 2940
19.0 1.3 155.5 3950
20.0 1.3 169.0 5200
17.0 1.4 125.5 2270
18.0 1.4 136.5 2980
19.0 1.4 148.5 3990
20.0 1.4 161.0 5260|
```

The interpolation of shaft power "PS" is in a sense performed in the 'inward out' direction:
first PS(and idem NS) on PDRA and then PS(and idem NS) for fixed PDRA on VS.
The expressions are the following:

PShaft = LEASQ(1, 3, "VS", "PDRA", "PS", VShip, Pitch, 3, 2, 1)
NShaft = LEASQ(1, 3, "VS", "PDRA", "N" ,VShip, Pitch, 3, 2, 1)

in which 1 is the Telitab set number, 3 is the number of address identifiers (Ndim%), the values for "Vs" and "PDRA" are given as interpolation input and "PS"/"N" (third address identifier) are the output values of the example functions. The selected order of the approximating curve in "VS" direction is 3 and in "PDRA" direction 2 is selected. '1' is added as value for Xtrap%, so no warning is issued in the event of extrapolation. The matrix interpolation can be applied in all except Method 2, obviously since only x,y data points are provided in that case.

NOTE: The matrix should be 'full', which means that the number of variation in each other should be constant. In the above table "VS" has 4 values in each of the 5 values for "PDRA".

Syntax 2: direct definition

The syntax is

LEASQ(0, Npoints%, x_1, y_1, x_2, y_2,…, x_n, y_n, xint, Order%, [Xtrap%=0,1])

If Pno%=0 (the first parameter in the expression), then all x_i and y_i values should be numeric expressions. The minimum number of x,y data points in the list Npoints%, being of X,Y points of the curve, is 2. Interpolation (and extrapolation) is performed according to the curve form selected in Order%.

Syntax 3 and 4: TeLiTab in object or string

These methods are similar to syntax 1. The data used is however not presented in the dataslot of the relation, but exists in an object or a telitab.

Quick links: Functions overview | Attribute overview | Constants overview | Dimensions overview