Introduction to Mathcad 15 – part 3

Statistics and data analysis in Mathcad

Mathcad allows to perform not only some basic statistical and data analysis operations, but also some data fitting interpolations. Statistical built-in functions are shown in the table below:

command	description	command	description
mean (A)	calculates mean of the elements of vector A	histogram(int,A)	Returns a matrix with frequencies as in hist(int,A) + the midpoints of the intervals
gmean(A)	Geometric mean of the elements of vector A	stdev(A)	Returns standard deviation of a given vector
hmean(A)	Harmonic mean of the elements of vector A	var(A)	Returns variance of given vector
kurt(A)	Kurtosis of the elements of vector A	corr(A,B)	Returns correlation coefficient between two given arrays
median(A)	median of the elements of vector A		
mode(A)	mode of the elements of vector A		
hist(int,A)	Returns a **vector** of frequencies with which the values fall in the intervals (int=number of intervals		

Mathcad also includesfour types of functions for working with probability denstities. Their names consist of two parts: letter defining function type and the name (or shortcut) of probability distribution, for example *dnorm(x,a,b)* – uses the normal distribution with the given parameters (a=mean, b=std. deviation)

- Probality densities – (letter d) gives a likelihood that a variable will take on particular value

- Distributions – (letter p) gives the probability that the variable will be less than or equal to specified value. Which is in fact the distribution in specified point.
- Inverse distribution – inverts the above function and for a given probability *p* gives the value such that random variable will be less than or equal to that value p.
- Random number generator – (letter r) generates a vector of m elements drawn from specified distribution.

Example:

Generate a vector of a 100 numbers based on normal(norm) and chi-square(chisq) distribution. Calculate basic statistics for this data. Plot a histogram with 10 intervals for both sets of data.

Data fitting

Besides doing statistical operation Mathcad is also capable of doing some data fitting calculations. In general they can be divided in two groups: regression and interpolation. As a result interpolation functions return curves that pass through all given points. So if the data is noisy it is better to use regression methods.

Interpolation:

- ***linterp(vx,vy,x)*** – links data points with linear segments. Vector **vx** must be defined in ascending order,
- ***lspline(vx,vy)*** – interpolates data with cubic splines with linear endpoints,
- ***bspline(vx,vy,y,n)*** – returns a vector of coeff. of B-spline which is used in *interp* function,
- ***cspline(vx,vy)*** – cubic spline with cubic endpoints,
- ***pspline(vx,vy)*** – cubic spline with parabolic endpoints,
- ***interp(vs,vx,vy,x)*** – returns interpolated y value corresponfing to x. Vector vs comes from one of the above spline functions.

Regression:

To obtain simple linear regression in Mathcad you have to create the line equation "by yourself" using two Mathcad comments:

- **slope(vx,vy)** – returns the slope(coefficient *a*) for the set of data
- **intercept(vx,vy)** – returns the y-intercept (coefficient *b*)

Then define the regression line and present it on the plot together with the set of data.

Using command ***line(vx,vy)*** returns both intercept and slope values in one move".

Mathcad also offers some more specialized regression models:

- **regress(vx,vy,n)** – returns coefficients for n-th degree polynomial that best fits the given set of data
- **loess(vx,vy,span)** – performs kind of local regression in the neighborhoods of data points (span defines how big the neighborhood >0). Returned vector can be used as an input to *interp* function presented above.

Non polynomial regression types require to define additional vector with guess values for the searched parameters. In the examples below this vector will be named *vg*. As a result all commands return a vector with parameters a, b, & c

- **expfit(vx,vy,vg)** – exponential curve given by the equation *a*exp(b*x)+c*
- **lgsfit(vx,vy,vg)** – logistic curve *a/(1+b*exp(-cx))*
- **logfit(vx,vy,vg)** – logarithmic curve *a*ln(x+b)+c*
- **pwrfit(vx,vy,vg)** – power curve $a*x^b+c$
- **sinfit(vx,vy,vg)** – sine curve *a*sin(x+b)+c*

You can also specify your own type of regression function and Mathcad will calculate its parameters in the same way as in the examples above.

To do this use one of the following commands:

- **linfit(vx,vy,F)** – generates the vector with parameters to create linear combination of the functions given as F (this must be specified before the command linfit)
- **genfit(vx,vy,vg,F)** – is a more generalized model using possibly nonlinear functions to fit the data.

It is more convenient to use the ***linfit*** function whenever it is possible, because the calculations are easier and it does not need a vector of guess values.

Some sets of data, especially coming from digital analyzers, can be disturbed by noise produced by analyzing equipment. For such sets of data some smoothing techniques coul be useful. Mathcad offers three built-in functions:

- **medsmooth(vy,n) –** uses the running median methods (n must be an odd number lower than number of elements in vy)

- **ksmooth(vx,vy,b)** – uses Gaussian kernel method to calculate the weighted averages ffor vy elements (use b which is few times bigger than the spacing between vx elements)
- **supsmooth(vx,vy)** – uses linear fitting for k-nearest neighbors (k is automatically taken) and returns the vector of fitted elements.

Exercise

1. Removing VOC from flue gases is carried on using activated carbon. In the table below you can find the experimental data[1] showing the results of adsorption of hexane and cyclohexane from air. The author claimed that the adsorption can be described with Langmuir isotherm:

$$q_{VOC} = \frac{abc_{VOC}}{1 + bc_{VOC}}$$

Where:

a – maximum adsorption volume

b – affinity parameter

c – concentration [ppm]

$q_{voc}=m_{voc}/m_{gac}$

C[ppm]	Temp [°C]	q_{voc}				
		33.6	**41.5**	**57.4**	**76.4**	**99**
200		0,080	0,069	0,052	0,042	0,027
500		0,093	0,083	0,072	0,056	0,042
1000		0,101	0,088	0,076	0,063	0,045
2000		0,105	0,092	0,083	0,068	0,052
3000		0,112	0,102	0,087	0.072	0,058

a. Using linear regression $1/q_{voc}$ vs $1/c_{voc}$ decide if the data are fitted well or not. Calculate parameters a and b.
b. Parameters a and b can be bound to the value of enthalpy of adsorption $\Delta_{ad}H$ and difference between activation energy of adsorption and desorption $\Delta_{b}H$ by Arrhenius type equations:

[1] M.S. Chou, J.H. Chiou (J. Envir. Engrg. ASCE,123, 437(1997)

$$a = k_a e^{-\Delta_{ad} H/RT} \quad b = k_b e^{-\Delta_b H/RT}$$

Check the fitting of data to those two equations and calculate the values of k_a, k_b, $\Delta_{ad}H$, $\Delta_b H$.

2. File "Exercise_03.xlmx" contains a set of data from gasification process. Import this set to your Mathcad file and prepare kinetic curves for all four components presented in the file.
   a. Smooth the raw the data to minimize the noise,
   b. Create a plot showing the change of volume in time for all four components. To this end you have to differentiate vectors of volume (dV/dt).

TIP:

To discretize the process of differentiation you can use following equations:

$$dX = \frac{X_n + X_{n+1}}{2} \qquad dY = \frac{Y_{n+1} - Y_n}{X_{n+1} - X_n}$$