1 Help on meshgrid

You can use the meshgrid command to generate two arrays containing the x- and y-coordinates at each position in a rectilinear grid. For example \([X,Y] = \texttt{meshgrid}(-5:1:5)\) returns two \(11 \times 11\) matrices - the X matrix defines the x-coordinates and the Y matrix the y-coordinates at each position in an \(11 \times 11\) grid. Try typing this command and view the contents of \(X\) and \(Y\).

This command is sometimes convenient for computing functions of 2 variables over a rectangular region of the coordinate system. For example, if we wanted to numerically compute the function \(f(x, y) = xy\) over a set of points in the range \(x \in [-5, 5]\) and \(y \in [-5, 5]\), this could be accomplished using element-wise operations on our meshgrid matrices by the command \(Z = X \cdot Y\). This is much simpler than cycling through each point in the space \([-5, 5] \times [-5, 5]\) to compute each point of the function.

For some standard operations such as multiplication, division, and power, element-wise operations are specified by a period in front of the standard operator: \(Z=\texttt{X.*Y}, Z=\texttt{X./Y}, Z=\texttt{X.^Y}\). This shouldn’t be confused with \(Z=\texttt{X*Y}\) which is interpreted as a matrix multiplication. Most other built-in math operations such as \(\sin(x)\), \(\cos(x)\) and \(\exp(x)\) are naturally element-wise.

Example: Say you want to generate a sampled representation of the 2-D sinusoid \(f(x, y) = x \sin(xy)\) in the range \(x \in [-5, 5]\) and \(y \in [-10, 10]\). This could be accomplished using with the following

\[
[x,y]=\texttt{meshgrid}(-5:1:5,-10:1:10);
\]

\[
z=x.*\sin(x.*y);
\]

A finer sampling of the function can be obtained by decreasing the step size, for example using \(\texttt{meshgrid}(-5:.2:5,-10:.2:10)\).

You can then display the function by typing \(z\) (return) or by using the \texttt{mesh} or the \texttt{image} command.

2 MATLAB Help on meshgrid

\texttt{MESHGRID} \hspace{1cm} X and Y arrays for 3-D plots.
\[
[X,Y] = \texttt{MESHGRID}(x,y)\hspace{2cm}\text{transforms the domain specified by vectors x and y into arrays X and Y that can be used for the evaluation of functions of two variables and 3-D surface plots.}
\]

\[
\text{The rows of the output array X are copies of the vector x and the columns of the output array Y are copies of the vector y.}
\]

\[
[X,Y] = \texttt{MESHGRID}(x)\hspace{2cm}\text{is an abbreviation for } [X,Y] = \texttt{MESHGRID}(x,x).
\]

Questions or comments concerning this laboratory should be directed to Prof. Charles A. Bouman, School of Electrical and Computer Engineering, Purdue University, West Lafayette IN 47907; (765) 494-0340; bouman@ecn.purdue.edu
[X,Y,Z] = MESHGRID(x,y,z) produces 3-D arrays that can be used to evaluate functions of three variables and 3-D volumetric plots.

For example, to evaluate the function \( x \exp(-x^2-y^2) \) over the range \(-2 < x < 2, -2 < y < 2\),

\[
[X,Y] = \text{meshgrid}(-2:.2:2, -2:.2:2);
Z = X .* \exp(-X.^2 - Y.^2);
surf(X,Y,Z)
\]

MESHGRID is like NDGRID except that the order of the first two input and output arguments are switched (i.e., \([X,Y,Z] = \text{MESHGRID}(x,y,z)\) produces the same result as \([Y,X,Z] = \text{NDGRID}(y,x,z)\)). Because of this, MESHGRID is better suited to problems in cartesian space, while NDGRID is better suited to N-D problems that aren’t spatially based. MESHGRID is also limited to 2-D or 3-D.

Class support for inputs X,Y,Z:
float: double, single

See also surf, slice, ndgrid.