MATLAB IV: Graphics with MATLAB

We begin by describing how to draw surfaces with MATLAB. (The commands plot and fplot are used to graph functions of one-variable. We will not discuss this.) The command to draw a surface is surf(x,y,z). Here x,y,z are matrices of the same size. Each triple of numbers taken from the corresponding positions in the matrices is one point on the surface.

Suppose that we want to graph \( f(x,y) = e^{-x^2 - y^2} \) over the square \(-2 \leq x \leq 2, -2 \leq y \leq 2\). The first thing to do is form matrices x,y whose entries are points in this square. We begin by forming a vector with equally spaced elements in the x-direction. The following command picks out points spaced 0.2 units apart between -2 and 2: \( xx=[-2:0.2:2] \); (the square brackets are not strictly necessary.) Next, do the same for the y direction, and merge them together with meshgrid.

\[
\begin{align*}
&>> \text{xx}=\text{-2:0.2:2}; \quad \text{yy}=\text{xx}; \\
&>> [x,y]=\text{meshgrid}(\text{xx},\text{yy}); \\
&>> z=\text{exp}(\text{-x}^2-\text{y}^2); \\
\end{align*}
\]

Notice that we did NOT write -x^2-y^2. We must put the dot before the ^ so MATLAB knows we want to square the ELEMENTS of the matrix rather than square the matrix itself (as in matrix multiplication). The same rule works for multiplication. \( x*y \) is different from \( x.*y \). The first one means multiply the matrices; the second one means multiply the corresponding elements of the matrices.

Finally, we use the surf command. We also label the x and y axes, give the graph a title, and print it. (This print command writes the figure as an encapsulated postscript file that can be sent to a printer using the unix command lpr. On many machines you don’t need to do this – you can print using a pull-down menu.)

\[
\begin{align*}
&>> \text{surf}(x,y,z) \\
&>> \text{xlabel(’x’)} \\
&>> \text{ylabel(’y’)} \\
&>> \text{title(’this is } f(x,y)=\text{exp(-x}^2-\text{y}^2)’)} \\
&>> \text{print -depsc fig-m11} \\
\end{align*}
\]
The MATLAB command for drawing level curves is `contour(x,y,z)`. Here \( x, y, z \) are just as for `surf`. With the values from before, we get level curves for \( f(x,y) = e^{-x^2-y^2} \) by typing

\[
\begin{align*}
&\gg \texttt{contour(x,y,z)} \\
&\gg \texttt{title('Level Curves for } f(x,y)=\exp(-x.\,^2-y.\,^2)')
\end{align*}
\]

`contour` has an optional fourth argument that controls how many level curves are drawn. For example, `contour(x,y,z,20)` forces MATLAB to draw level curves corresponding to 20 \( z \) values.

Let’s draw level curves for \( f(x,y) = \sin(\pi x/4)\cos(\pi y/4) \).

\[
\begin{align*}
&\gg \texttt{uu=-4:0.2:4; vv=uu; [u,v]=meshgrid(uu,vv);} \\
&\gg \texttt{w=\sin(pi*u/4).*\cos(pi*v/4);} \\
&\gg \texttt{contour(u,v,w,20)}
\end{align*}
\]

`\( \sin \)` and `\( \cos \)` are built-in functions that operate on the entries of a matrix. Also, we used \( .* \) between two matrices but just \( * \) and \( / \)
when multiplying by $\pi$ and dividing by 4 ($\pi$ is a constant built-in to MATLAB).

To draw vector fields with MATLAB, use the command \texttt{quiver}. This command takes four matrices as arguments, all of the same size. \texttt{quiver(x,y,u,v)} draws vectors determined by $u,v$ at the points $x,y$.

For example, suppose we wish to sketch the vector field $f(x,y) = (1-x, -y)$ on the square $-3 \leq x \leq 3$, $-3 \leq y \leq 3$.

\begin{verbatim}
>> xx=-3:0.3:3; yy=xx; [x,y]=meshgrid(xx,yy);
>> quiver(x,y,1-x,-y)
\end{verbatim}

Notice that when we write $1-x$, MATLAB assumes that we mean to do the subtraction on each entry. Also notice that our meshgrid is spaced 0.3 here. It is generally best to use a coarser grid with \texttt{quiver} than with \texttt{surf} or \texttt{contour}.

\begin{verbatim}
>> quiver(x,y,1-x,-y)
\end{verbatim}

\texttt{quiver} has an optional fifth argument that scales each of the arrows. For example, \texttt{quiver(x,y,u,v,2)} makes each arrow twice as long as \texttt{quiver(x,y,u,v)}. However, making the arrows longer usually makes them overlap which looks a little strange.
Let’s do one final example of drawing vector fields. Let’s draw $f(x,y) = (x^2y, 2)$ on the square $-1.5 \leq x \leq 1.5, -1.5 \leq y \leq 1.5$. We might be tempted to try

```matlab
>> xx=-1.5:0.25:1.5; yy=xx; [x,y]=meshgrid(xx,yy);
>> quiver(x,y,x.^2.*y,2)
??? Error using ==> quiver
Matrix C must be the same size as Z.
```

The problem here is that MATLAB doesn’t understand the 2-`quiver` accepts only matrices. One (hack) way to avoid this problem is to write `quiver(x,y,x.^2.*y,2+x-x)`. A more elegant solution is to use the command `ones`. This command creates a matrix of whatever size we want filled entirely with 1’s. To get the correct size, we can use the command `size`.

```matlab
>> xx=-1.5:0.25:1.5; yy=xx; [x,y]=meshgrid(xx,yy);
>> quiver(x,y,x.^2.*y,2*ones(size(x)))
```