Animated Graphs in MATLAB

There are many occasions where the response of a system under investigation changes over time or space. To illustrate the temporal or spatial variability, a sequence of static images (graphs, figures) are created for a number of different points in time or space. While presenting those static images is fine for a printed document, presenting an animated (movie) of the sequence of images can greatly enhance a "live" presentation. While a number of software tools are available to create these animated presentations, one of the easiest to use is MATLAB.

Creating an animated sequence of images in MATLAB requires a bit of planning in generating the data that will be used to construct the individual images (or frames) of the animated sequence. One easy way is to have all of the numerical data that is to be displayed placed in a single data file (usually created by some simulation program). The data is then loaded into a single MATLAB matrix, and the data plots are generated frame-by-frame in a type of `do loop`. As each plot is generated, the image is "captured" as a frame of the movie. After all of the frames are captured, they can be played back at any speed (frames/second) and as many times as the user desires. If the workspace is saved after generating the frames, it can be reloaded later and the movie replayed without having to regenerate the images.

The following example illustrates this technique using MATLAB to produce 2 and 3-D images of groundwater head in a groundwater aquifer. Suppose that the groundwater basin is defined by 31 nodes in the x and y direction. For each timestep in a groundwater simulation model, the hydraulic head is calculated at each of the 31 × 31 nodes. Further suppose that the simulation model saved the results in a single file called `gwhead.out` by writing out the hydraulic heads at each node at the end of 21 different time periods. The output file would then contain 31 numbers/line, and 31 lines/time period (or a total of $31 \times 21 = 651$ lines). The program will also write out a file called `gwtimes.out` that contains the time associated with each time period that is to plotted later in MATLAB. This file contains one number/line, and the values will serve as part of a label on each frame of the movie. The following MATLAB commands will generate the movie frames.

```
nrow=31
ntimes=21  %number of time sequences (or frames)
nrep=5 %number of times to play the movie
fps=3 %frames/sec to play back the movie
load gwhead.out  %reads datafile and stores values in matrix gwhead
load gwtimes.out %read datafile containing time labels for frames
figure('position',[100,100,800,600]) %left, bottom, width, height in pixels
for k=1:ntimes
  surf(gwhead((k-1)*nrow+1:k*nrow,:))
  view([-100,30])  %sets rotation and angle of viewing the surface plot
  axis([1,31,1,31,43,50])  %set x, y, and z axis minimum and maximum values
  text(3,30,44,['Time: ',int2str(gwtimes(k)),' (days)']) %put label on frame
  m(k)=getframe; %captures the kth frame in the matrix m
end
```

At this point, all of the frames of the movie would be generated and stored in matrix `m`. If the workspace is saved, it can be loaded later and the movie available for viewing without having to re-generate the frames.

To play back the movie, use the command

```
movie(m,nrep,fps)  %play back the movie stored in matrix m
```

A similar procedure can be used to generate 2-D plots of the hydraulic head along a row or column of the solution grid. For example, suppose we wanted to show the hydraulic heads along column 15 of the grid for each of the 20 time periods. First generate the frames of the movie using the following commands.

```
x=linspace(0,3000,31)  %generate the x axis values for each grid point
for k=1:ntimes
  plot(x,gwhead((k-1)*nrow+1:k*nrow,15))  %x-y plot of head along column 15
  axis([0,3000,43,50])  %set x and y axis minimum and maximum value
  text(2000,49,['Time: ',int2str(gwtimes(k)),' (days)']) %put label on frame
  n(k)=getframe;  %capture the kth frame in the matrix n
end
```

To play back this movie, use the command

```
movie(n,nrep,fps)  %play back the movie stored in matrix n
```