**Animation in Matlab**

**For Summer 1997 [Envision-It!](http://physics.gac.edu/%7Ehuber/envision/) Workshop**

One of the uses for Matlab is to make a series of calculations and plots.  For example, in the   
 [concurrent/countercurrent flow](http://physics.gac.edu/%7Ehuber/envision/flow/welcome.htm) model, we wanted to calculate the amount of material after each time iteration and plot this on the screen.  To do this, we had the program difflow.m

% DIFFLOW.M   Concurrent Flow model   
% This script models the flow of material from a "source" stream to a   
% "sink" stream which are separated by a semi-permeable membrane.   
% In the concurrent flow case, these two streams are traveling in   
% the same direction, and in the countercurrent case, they   
% are traveling in opposite direction.   
%   
% Tom Huber, July 1997   
%

ncells = 25;        % Number of Cells in the Array   
NTimes = 50;        % Number of Time steps to perform

MSrce = zeros(ncells,1);  % Initialize the Array to have zero concentration   
MSink = MSrce;             % Set up second array   
VMax = 100;                % Maximum Concentration   
VSrce = 1;                 % Volume Flow rate of Source Stream   
VSink = 1;                 % Volume Flow rate of Sink Stream   
k = .1;                    % Diffusion Constant

MSrce(1) = VMax;           % Set initial cell for source stream

for i=1:NTimes  % Perform a total of NTimes time steps   
  dm = k\*(MSrce/VSrce - MSink/VSink);   
  MSrce = MSrce - dm;   
  MSink = MSink + dm;   
  MSrce = [VMax; MSrce(1:ncells-1)];   
  MSink = [0 ; MSink(1:ncells-1)];  % Concurrent flow case   
  clf   
  plot (MSrce,'r-')   
  hold on   
  plot (MSink,'g-')   
  legend('Source Stream','Sink Stream')   
  xlabel('Location')   
  ylabel('Mass Content')   
  title(['Time Step ' num2str(i)])   
  drawnow   
end  % for i=1...

The following code performs the same calculation, but updates the graphs much faster than the original version.

%<pre>   
% DIFFANIM.M   Concurrent Flow model   
% This script models the flow of material from a "source" stream to a   
% "sink" stream which are separated by a semi-permeable membrane.   
% In the concurrent flow case, these two streams are traveling in   
% the same direction, and in the countercurrent case, they   
% are traveling in opposite direction.   
%   
% Tom Huber, July 1997   
%   
% Demonstrates how to do Animation in Matlab

ncells = 25;        % Number of Cells in the Array   
NTimes = 50;        % Number of Time steps to perform

MSrce = zeros(ncells,1);  % Initialize the Array to have zero concentration   
MSink = MSrce;             % Set up second array   
VMax = 100;                % Maximum Concentration   
VSrce = 1;                 % Volume Flow rate of Source Stream   
VSink = 1;                 % Volume Flow rate of Sink Stream   
k = .1;                    % Diffusion Constant

MSrce(1) = VMax;           % Set initial cell for source stream

clf   
HandleSrce = plot (MSrce,'r-');  % Keep the handle for this plot   
hold on   
HandleSink = plot (MSink,'g-');   
HandleTitle = title('Time Step 0');  % Keep the handle for the title   
legend('Source Stream','Sink Stream')   
xlabel('Location')   
ylabel('Mass Content')   
set (HandleSrce,'Erase','xor')  % Set these so it updates   
set (HandleSink,'Erase','xor')   
set (HandleTitle,'Erase','xor')

for i=1:NTimes  % Perform a total of NTimes time steps   
  dm = k\*(MSrce/VSrce - MSink/VSink);   
  MSrce = MSrce - dm;   
  MSink = MSink + dm;   
  MSrce = [VMax; MSrce(1:ncells-1)];   
%  MSink = [0 ; MSink(1:ncells-1)];  % Concurrent flow case   
  MSink = [MSink(2:ncells) ; 0];  % Countercurrent flow case   
  set(HandleSrce,'YData',MSrce)   % Change just the Y Data for plot   
  set(HandleSink,'YData',MSink)   
  titlestring = ['Time Step ' num2str(i)]   
  set(HandleTitle,'String',titlestring)  % Change just the title   
  drawnow   
end  % for i=1...

In Matlab, all graphics objects have a "Handle" which uniquely identifies the object.  The statement

HandleSrce = plot (MSrce,'r-');

not only draws the plot, but stores the handle for this graphics object in the variable HandleSrce. At a later point in the script file, we can use this handle to modify other properties of this graph.  To allow us to update just the Y data on the graph, we need to execute the command

set (HandleSrce,'Erase','xor')

which instructs Matlab how to erase the line.  Now the real trick in animation comes with the   
statement

set(HandleSrce,'YData',MSrce)

This changes just the Y values for the graph with handle HandleSrce.  Using this method, the   
program just redraws the line without redrawing the coordinate system, title, legend, etc.  This is why this method is so much faster than doing a plot each time.

Electronic Copy: http://physics.gac.edu/~huber/envision/matgui/animate.html   
Created: 7-JUL-97 by [Tom Huber](http://physics.gac.edu/%7Ehuber/), [Physics Department](http://physics.gac.edu/), [Gustavus Adolphus College](http://www.gac.edu/).