**How to Make a Movie from Matlab**

This page can be reached via my homepage at [http://www.math.umbc.edu/~gobbert](http://www.math.umbc.edu/%7Egobbert).

**Acknowledgement**

I have to thank Kathleen Hoffman for providing me originally with this solution to my long-standing problem of how to create movies from Matlab plots. I have only summarized the information here for the benefit of the interested community.

**Purpose of this Document**

I receive with a certain regularity the question: ``How do I make movies with Matlab?'' In fact, that was one of my burning questions some while back myself. The problem is that there is no easy answer that satisfies all relevant requirements, see one paragraph below.

First of all, Matlab does have a movie feature. But is has several severe drawbacks: It stores *all* movie frames in memory, it has to be loaded before running (i.e., always runs one more time than you really desire), and it does not provide a `web-ready' format. These problems severely restrict the usefulness of Matlab's movie feature. (I checked this out thoroughly using Matlab 5, so I do not know whether anything improved in Matlab 6 (R12).)

What is the real reason to use movies in science and engineering? If the solution is three-dimensional and time-dependent in nature, then a movie is often the only sensible way to look at the entire sequence of results. In this case, it is not desirable and often impossible to store more data than necessary for one frame in memory.

The other common reason for making movies is that one wants to jazz up a presentation or a webpage. Therefore, it is important to be able to produce the final movie in some standard format that can be understood on the web.

Therefore finally, this note will present the solution in two steps: (1) First, Matlab is used as the graphing engine to produce *each* frame of the desired movie individually and save the result to a file. (2) Then, the Linux utility mpeg\_encode is used to concatenate the individual movie frames into one mpeg-file.

This procedure satisfied my requirements. If you are interested in using a different package than Matlab, you just have to replace the first step.

**How to Make the Individual Frames**

WARNING: The drivermakemovie.m file currently provided here as example has NOT actually been tested by me, yet! So, it likely not produce sensible picutures, if it works at all. Please, have patience until I get back to Baltimore in August.

I usually use a driver file like [drivermakemovie.m](http://www.math.umbc.edu/%7Egobbert/drivermakemovie.m) to automate the creation and export of each frame. To start, create a directory called ppmmovie, where the movie frames will be stored; I only advise this, because I do not want to clutter up my current directory unnecessarily.

The example [drivermakemovie.m](http://www.math.umbc.edu/%7Egobbert/drivermakemovie.m) enclosed here creates the plot directly from a given formula. In practice, it is more likely that this will be again be split up into the two steps of reading result data from a file and creating the plot; that's what I use a code called makemovie.m for, which explains the name for drivermakemovie.m.

Basically, drivermakemovie runs through a time loop. For each time, a plot is created and printed to a ppm-file, which will from one movie frame. It is important to note how the filename is chosen: Its first part is the directory name ppmmovie, then a basename movie is used, then the loop counter n attached, and finally the extension chosen. The counter n is crucial and must be continuously counted integers, i.e., step size 1 in the loop on n is required. The basename movie is not crucial as such, but as the files will only be temporary, there is no reason to choose anything different.

A few additional, typically useful commands are used: The color limits are set to the same limits for each plot; only in this way will colors have the same meaning for different movie frames. Each plot is given a title that lets the viewer distinguish the frames later.

After running drivermakemovie, you should have the files movie0000.ppm, movie0001.ppm, ..., movie0016.ppm in the directory ppmmovie.

**How to Concatenate the Individual Frames to a Movie**

Copy the file [paramfile](http://www.math.umbc.edu/%7Egobbert/paramfile) into the directory ppmmovie. Change directory to ppmmovie. Make sure you have the files of the form movie\*.ppm.

In the file paramfile, you only have to change very little, whenever you use it. The line ``INPUT movie\*.ppm [0000-0016]'' specifies the name of the input files (the individual movie frames) to be movie\*.ppm with the ``\*'' replaced successively by 0000, 0001, ..., 0016. The range of numbers in this line is the only thing I ever change.

Finally, create the mpeg-movie by the Linux utility mpeg\_encode using the information from paramfile by issuing the following command at the Linux prompt mpeg\_encode paramfile This should create a file called movie.mpeg in the directory ppmmovie.

The line ``OUTPUT movie.mpeg'' in paramfile chooses the name of the output file. I never change this, but rather rename the file later to any desired name using the Unix command mv.

Remember to delete the ppm-files after creating the movie, because they are now useless and very large in real applications. Since those files were only used temporarily, I suggested to stick with the basename movie above.

**How to Watch an mpeg-Movie**

There are, of course, many possibilities on the various platforms, which is exactly, why we chose the mpeg-format for our movie. On Linux, I have found that at least one of the utilities xanim or mtv is usually available. Simply call them followed by the filename of the movie, e.g., movie.mpeg.

To run a movie directly from a webpage, it might be necessary first to set the preferences in your browser to react properly to react to a mpeg-file in a href-link. Talk to your local web guru for more information on setting up a browser.

Copyright © 2001 by Matthias K. Gobbert. All Rights Reserved.   
This page version 0.6, June 2001.