

# Computer Animations



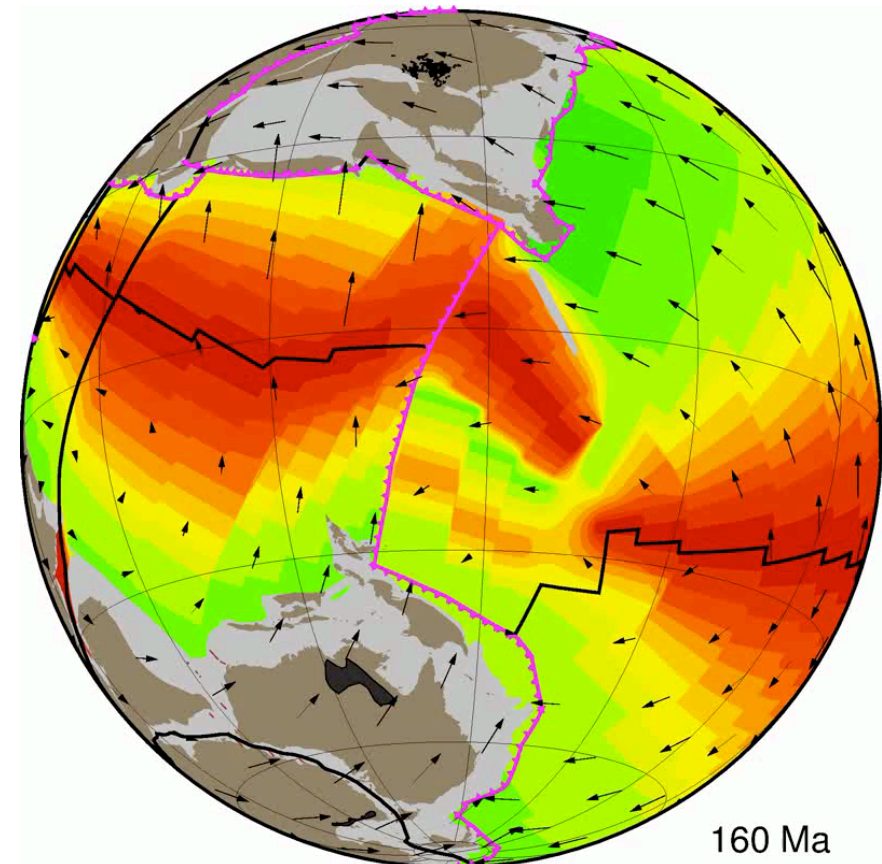
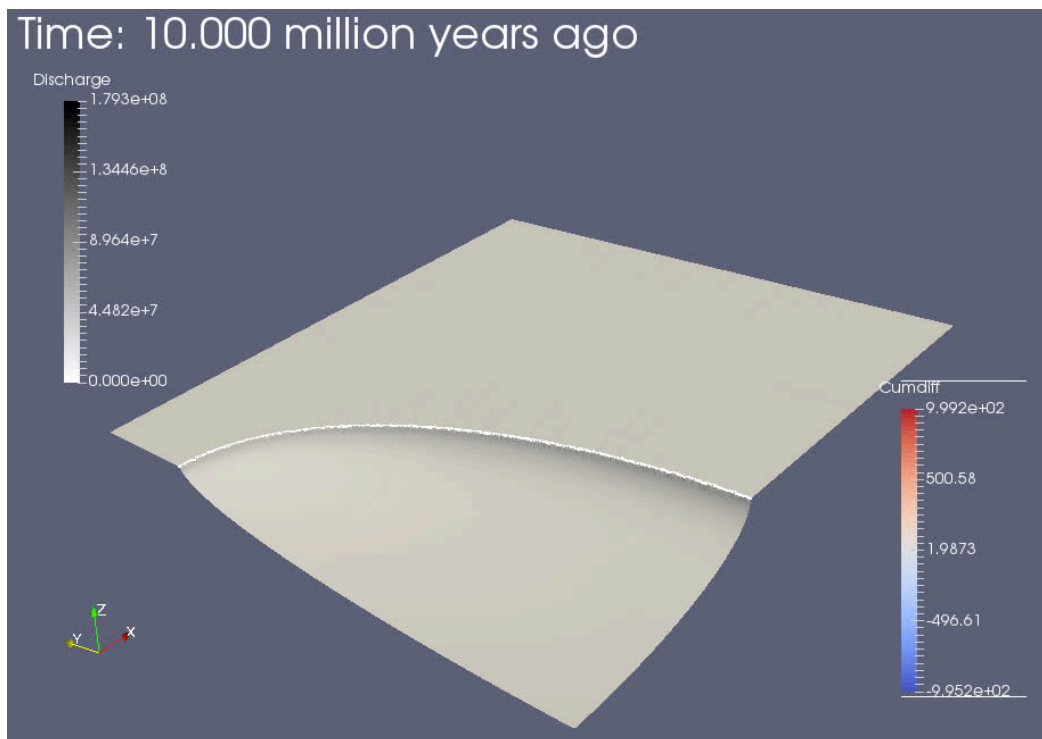
- We will cover the steps typically needed to generate an animation with GMT:
  - The first hurdle is to make all PostScript frames with GMT and convert to JPGs or similar raster images
  - The second is to convert these to a movie; this step is device-dependent, with different solutions for OSX, Windows, and Linux

# Computer Animations

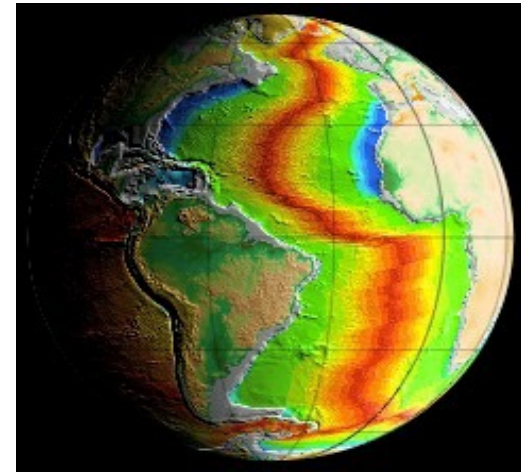
● What is computer animation?

● A rapid display of still images (frames)

● Delay between frames and change in content determine the smoothness of your movie



# Computer Animations



- How will we do it?
  - bash script with a loop over “time” or “frame”
  - Use GMT to make a still image for each new time/frame
  - Convert PostScript plots to raster images of given size/resolution
  - Display images in correct order
    - Build an animated GIF file (short clips only)
    - Build a MP4 movie (or other video formats)

# Installing ImageMagick and FFMPEG

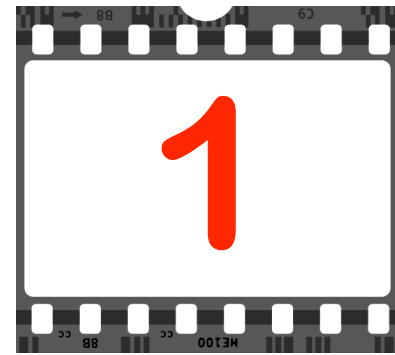
- Ensure that you have Xcode installed
- Install Macports
- To install ImageMagick  
`sudo port install ImageMagick`
- To install FFMPEG  
`sudo port install ffmpeg`

# Animation Script Steps



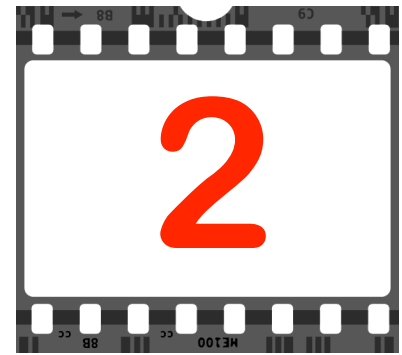
- All scripts have three main parts:
  - Initialization
  - Main frame loop generating the stills
  - Clean up and packaging

# Animation Step One



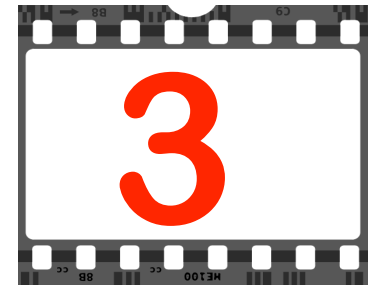
- Initialize control variables:
  - Set geometry and number of frames
  - Set frame names and sequence
  - Create subdirectory for images
  - Make temporary file names
- Perform initial (time-independent) calculations:
  - Get data and pre-processes them
  - Create static background map overlays
  - Make static cpt files or symbols

# Animation Step Two



- Set and execute the frame loop
  - Increment the frame counter
  - Increment auxiliary “time” variables
- Do time-dependent calculations
  - Process data for this  $t$
  - Create any other files needed for this  $t$
- Make the frame plot PostScript file
- Convert PostScript to raster image
  - Use ps2raster to create a JPG image
    - Give each image a unique and sequential name
    - Frame number starts at 0, and is padded with leading zeroes

# Animation Step Three



- Remove any temporary files and data sets created above
- Post-process the frame images:
  - Convert image frames to an animated GIF file
    - Use convert (ImageMagick) or similar tool (gifsicle)
  - Convert image frames to movie
    - Use ffmpeg or similar tool (QuickTime, iMovie, MovieMaker, Premiere, etc.)
- Give user instructions on how to play movie.



# Movie Clip Uses

- Include in PowerPoint or KeyNote presentations for class or conferences
- Include on your website
- Include in PDF documents [yes, you can]
- Include as dynamic content with some journals (e.g., AGU's JGR)
- Include in podcasts for education
- Build HD versions for documentaries

# Frame Name Details

- Place all final images in a single sub-directory
  - Use `mkdir -p` to create a directory (and subdirectories recursively)
- Use a frame counter variable in the loop
  - Let 0 be first frame and increment by 1
- Format number with enough leading zeros to ensure correct sequencing when viewing (example below would turn frame 5 into 000005)
  - `padded_frame=$(printf "%06d" $frame)`

# Rasterization of PostScript

- We use ps2raster (psconvert in GMT5) to convert a PostScript file to a JPG image (-Tj)
- \$dpi is resolution in pixels per inch
- -A clips to bounding box (if you have problems, check that the images have the same pixel dimensions)

```
ps2raster -Tj -E$dpi -A $$ps
```

# Build your movie

## 🍌 Convert JPG frames to animated GIF:

- ▶ `convert -delay rate -loop n dir/*.jpg movie.gif`
- ▶ Here, **rate** is in 1/100th of a second
- ▶ Loop **n** times (0 means forever)

## 🍌 Convert JPG frames to MP4 movie:

- ▶ `ffmpeg -r rate -b bitrate -i dir/movie_%06d.jpg\ movie.mp4`
- ▶ Here, **rate** is in frames per seconds [25]
- ▶ **bitrate** is in kbytes/sec [200]
- ▶ Assumes images are named `movie_#####.jpg`

These tools can be installed on most Unix-like systems. Cygwin users on Windows are better off doing movie-building with native Windows software

# Build your movie

## FFMPEG tips

- ▶ `"-vf scale=800:-1"` sets the width of the animation to be 800 pixels, and adjusts the height to maintain the correct aspect ratio

## Convert JPG frames to QuickTime MOV movie:

- ▶ `ffmpeg -framerate 5 -pattern_type sequence -i 'PNG_AgeGrid_vels_%06d.jpg' -vf scale=800:-1 -vcodec mpeg4 -b 2500k -r 5 out.mov`

**These tools can be installed on most Unix-like systems. Cygwin users on Windows are better off doing movie-building with native Windows software**

# Loops in BASH

```
frame=0 # Initialising frame
```

```
max__frame=10
```

```
while (( $frame <= $max__frame )); do
```

```
    padded__frame=$(printf "%06d" $frame)
```

```
    psfile=gmt__rox__${padded__frame}.ps
```

```
    [ All your plotting commands ]
```

```
    ps2raster -Tj -A ${psfile}
```

```
done
```

```
ffmpeg -pattern__type sequence -i
```

```
'gmt__rox__%06d.jpg' [ parameters ] outfile.mov
```

# Browser Video Issues

- Not all browsers support the <video> tags yet, and things are changing constantly [HTML5]
- Even when they do, they tend to support different codecs [why can't we all get along...]
- Find a format that works for your OS and browser
  - E.g., wmv for Windows
  - swf for Flash
  - mp4/m4v for Mac/iPad



# Class exercise

- 🌐 Create an animation of a spinning globe using whatever gridded dataset you like as the basemap
- 🌐 Tips:
  - 🌐 You may need to downsample your grids to make the plotting quicker
  - 🌐 You can experiment with illumination (sun angles), but this will slow down the process