Top 10 MATLAB tricks for beginners

Matthew Roughan

July 11, 2017

Tips

1. Everything (almost) in MATLAB is an array of *floating-point* numbers.

   You'd better know what that is, and what its limitations are.

   http://www.ee.columbia.edu/~marios/matlab/Fall96Cleve.pdf

2. Make your code neat. Trust me!

3. The `plot` command has a *lot* of options, *e.g.,* to control font size. Make sure your graphs are readable, or even dare I say it, beautiful.

4. Performance tips:

   - Pre-allocate arrays.
   - MATLAB is optimised for vector and matrix operations, not loops.
   - Keep loops tight (avoid unnecessary ops inside frequent loops).
   - Learn to measure time using `tic/toc` and to profile.

5. Functions I use again and again

   `max, min, sort, unique, cumsum, find, sum,`

6. MATLAB has a function for almost anything. Learn to use its built in tools to find them `help` and `lookfor` plus the online pages (with Google to search).

7. Don't hard code data into algorithms. Really, really, really don't do it twice for the same piece of data.

8. Don't use global variables.

9. Function names should match the file they are kept in.

10. The following will generate the Batman symbol.

```
xr = linspace(-7,7,1500);
yr = linspace(4.5,-4.5,1500);

x = repmat( xr , [ numel(yr) 1 ] );
y = repmat( yr' , [ 1 numel(xr) ] );

batman = (((x/7).^2.*sqrt(abs(abs(x)-3)./(abs(x)-3))+((y/3).^2) .*  ...
           sqrt(abs(y+(3*sqrt(33)/7))./(y+(3*sqrt(33)/7))))-1) .* ...
         (abs(x/2)-((3*sqrt(33)-7)/112).*(x.^2)-3+sqrt(1-(abs(abs(x)-2)-1).^2) - y) .* ...
         (9*sqrt(abs((abs(x)-1).*(abs(x)-0.75))./((1-abs(x)) .* (abs(x)-0.75)))-8*abs(x)-y) .* ...
         (3*abs(x)+0.75*sqrt(abs((abs(x)-0.75).*(abs(x)-0.5))./((0.75-abs(x)).*(abs(x)-0.5)))-y) .* ...
         (2.25*sqrt(abs((x-0.5).*(x+0.5))./((0.5-x).*(0.5+x)))-y) .* ...
         (((6*sqrt(10))/7)+(1.5-0.5*abs(x)) .* sqrt(abs(abs(x)-1)./(abs(x)-1))-((6*sqrt(10))/14).*sqrt(4-(abs(x)-1).^2)-y);

imagesc( log(abs(batman)) );
```

Top ten cute "cook-book" examples:

1. MATLAB has routines and shortcuts to generate all sorts of vectors and arrays. Try out

   ```
   x = [1:3:7], y = [10:-1:0], z = rand(3,3), k = zeros(2,3), g = eye(2)
   ```

2. Take every second element of a vector, *e.g.,*

   ```
   x(1:2:end)
   ```

3. Check a condition, *e.g.,*

   ```
    assert( x == y, 'Error: x should be equal to y')
   ```

4. Sort a pair of arrays in the order given by the first

   ```
   [x, i] = sort(x)
   y = y(i)
   ```

5. Calculate the mean of the values that are finite, *e.g.,*

   ```
   mean( x( isfinite(x) ))
   ```

6. Calculate L^2 norm of a vector, *e.g.,*

   ```
   sum( x.^2 )
   ```

7. Check that all of a vector satisfies a condition, *e.g.,*

   ```
   if all(x == 1)
     ...
   end
   ```

8. Check if any of a vector violates a condition, *e.g.,*

   ```
   if any(x ~= 1)
     ...
   end
   ```

9. Minimum or maximum of a 2D array, *e.g.,*

   ```
   min( min( A ) )
   ```

10. Find the last 3 elements of a vector, *e.g.,*

    ```
    x(end-2:end)
    ```

For more hints and links see

http://www.ee.columbia.edu/~marios/matlab/matlab_tricks.html