

**Matlab<sup>®</sup> Guide****for****Device Characterization and Circuit Design Project**

This handout gives a very brief overview of useful matlab<sup>®</sup> functions and code samples for handling weblab measurement data.

If you have serious matlab<sup>®</sup> trouble, send me an email with your code and error message(s). Simple syntax checking should be left to matlab<sup>®</sup>, please :)

**1. Matlab<sup>®</sup> functions**

General reference at

<http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.shtml>

Reference listing functions by category (**very useful!!**)

<http://www.mathworks.com/access/helpdesk/help/techdoc/ref/ref.shtml>

Functions you may find yourself using:

```
clf;  
subplot(...);  
plot(...);  
semilogy(...);  
hold on;  
axis(...);  
sum(...)  
xlabel(...);  
ylabel(...);  
title(...);  
floor(...)  
orient ...;
```

Important: for multiplication and division, use the element by element operators `.*` and `./`  
`*` and `/` are matrix operators, which you don't want to use.

## 2. Code Samples

### 2.1. Reading the Weblab data file

To read directly the CSV file from matlab<sup>®</sup> without having to erase the first 3 lines, create a function in a file called *readWeblabFile.m*, with the following content:

```
% returns a matrix from the weblab data-file <filename>
function [data]=readWeblabFile(filename)

fid=fopen(filename,'r');
data=[];
if (fid==-1)
    disp(['ERROR: file not found or could not be opened for read. check file ' filename]);
    break;
end
% skip first lines of comments
sLine=fgetl(fid);
sLine=fgetl(fid);
sLine=fgetl(fid);
% read the data
while 1
    sLine=fgetl(fid);
    if (~ischar(sLine))
        % end of file
        break;
    end
    % add line to table
    data=[data;str2num(sLine)];
end
fclose(fid);
```

### 2.2. Extracting data

Once you load the data, you must extract the columns that correspond to I, V, etc. Example: read mos.csv and take the first column to save in variable ID, 2<sup>nd</sup> column as VG, etc. For the correct order, look at the comment lines in the csv files.

```
MOS = readWeblabFile('mos.csv');
ID = MOS(:,1);
VG = MOS(:,2);
VD = MOS(:,3);
...
```

### 2.3. Derivatives

A cheap derivative is to take the difference of two neighboring data points. It won't be very smooth, but it's sufficient here. You can avoid a for loop if you use the matrix capabilities of matlab<sup>®</sup>. Here we take  $r_o = 1 / [ dI_D/dV_{DS} ]$

```
roMOS = 1./ ( ([0;ID]-[ID;0])./([-1;VD]-[VD;-1]) ); roMOS(1)=[];
```

What is being done? With  $([0;ID]-[ID;0])$  we shift ID and subtract it. This is  $I_D(n)-I_D(n+1) = dI_D$ . Similarly,  $dV_{DS}$  is computed. The quotient is  $g_o$ , and  $1/g_o = r_o$ . The second line,  $roMOS(1)=[]$ ; trims the derivative back to the original length, since we inserted an extra row to find the difference (we said  $[0;ID]$  etc). The choice of 0 and -1 in the equation is arbitrary, since the

boundary elements of the derivative are meaningless. The -1 was picked to avoid a divide by zero error.

Example: for  $ID=[1,4,9,16,25,36]$  and  $VD=[1,2,3,4,5,6]$  we get

row #	$[0;ID]$	$[ID;0]$	$dI =$ $[0;ID]-[ID;0]$	$[-1;VD]$	$[VD;-1]$	$dV =$ $[1;VD]-[VD;1]$	roMOS
1	0	1	-1	-1	1	-2	2
2	1	4	3	1	2	1	1/3
3	4	9	5	2	3	1	1/5
4	9	16	7	3	4	1	1/7
5	16	25	9	4	5	1	1/9
6	25	36	11	5	6	1	1/11
7	36	0	13	6	-1	5	-1/13

#### 2.4. Plotting data:

Since all the data is in one table, we cannot just plot

```
plot(VD,ID);
```

because we would end up with nasty lines drawn through the graph (try it out to see!) Instead we can split the data up, and plot ID vs VD for every different VG. We can do this automatically:

```
clf; % clear graph
vnum=sum([inf;VG]~=[VG;inf])-1; % count the number of VG's we have
vsize=floor(size(VG,1)/vnum); % for each VG there are this many data points of ID vs VD
% loop through the data, and plot for each VG:
for k=[0:vnum-1]
    idx=[(1+k*vsize):(k*vsize+vsize)-1]; % this is where the data is in the big table
    subplot(1,2,1); % 2 graphs on this page
    hold on; % we want to overlap graphs
    plot(-VD(idx),-ID(idx)); % I-V characteristics
    xlabel('VSD');ylabel('-ID'); % label it
    axis tight; % let's not waste figure space

    subplot(1,2,2); % next plot: ro vs. VSD
    hold on;
    plot(-VD(idx),roMOS(idx)); % plot ro vs VSD
    xlabel('VSD');ylabel('roMOS');
    axis([0,3,0,1e5]); % because the boundaries of the derivative may
    % be some oddly large number, fix the axis to only the range of interest.

    % add more code here
end
orient tall; % options for printing: use the whole page available. Looks less shabby!
```

Repeat this kind of loop for each file that you are processing. If you want to add more graphs, just add similar code where it says % add more code here You may want to adjust the subplot command to accommodate more graphs on one page.

The complete example:

```
% plot I-V and ro for mosfet measurement
MOS = readWeblabFile('mos.csv');
VG = MOS(:,1);
VD = MOS(:,2);
ID = MOS(:,3);

% take the derivative to get ro
roMOS = 1./([0;ID]-[ID;0])./([-1;VD]-[VD;-1]); roMOS(1)=[];

clf; % clear graph
vnum=sum([inf;VG]~=[VG;inf])-1; % count the number of VG's we have
vsize=floor(size(VG,1)/vnum); % for each VG there are this many data points of ID vs VD
% loop through the data, and plot for each VG:
for k=[0:vnum-1]
    idx=[(1+k*vsize):(k*vsize+vsize)-1]; % this is where the data is in the big table
    subplot(1,2,1); % 2 graphs on this page
    hold on; % we want to overlap graphs
    plot(-VD(idx),-ID(idx)); % I-V characteristics
    xlabel('VSD');ylabel('-ID'); % label it
    axis tight; % let's not waste figure space

    subplot(1,2,2); % next plot: ro vs. VSD
    hold on;
    plot(-VD(idx),roMOS(idx)); % plot ro vs VSD
    xlabel('VSD');ylabel('roMOS');
    axis([0,3,0,1e5]); % because the boundaries of the derivative may
    % be some oddly large number, fix the axis to only the range of interest.

    % add more code here

end
orient tall; % options for printing: use the whole page available. Looks less shabby!
```