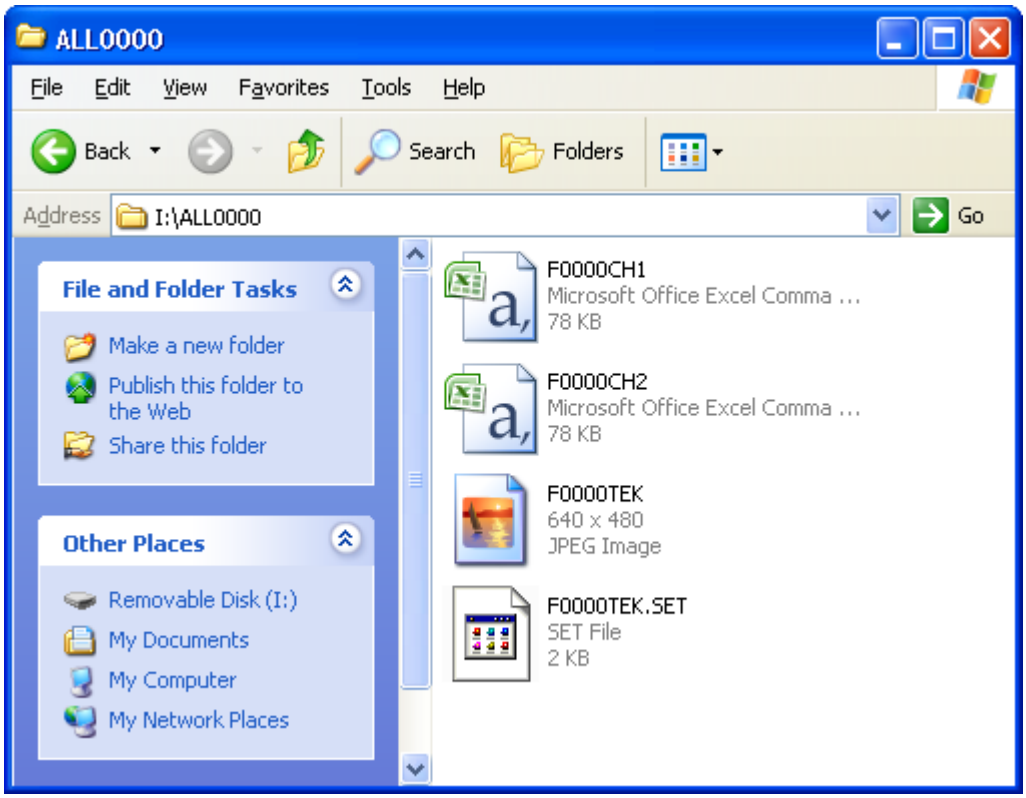
Analysis of data from CD4007 XY plots

Additions to lab procedure:

When you do a “Save All to File”, you end up with a folder on the USB stick that has a separate .csv file for each scope trace:



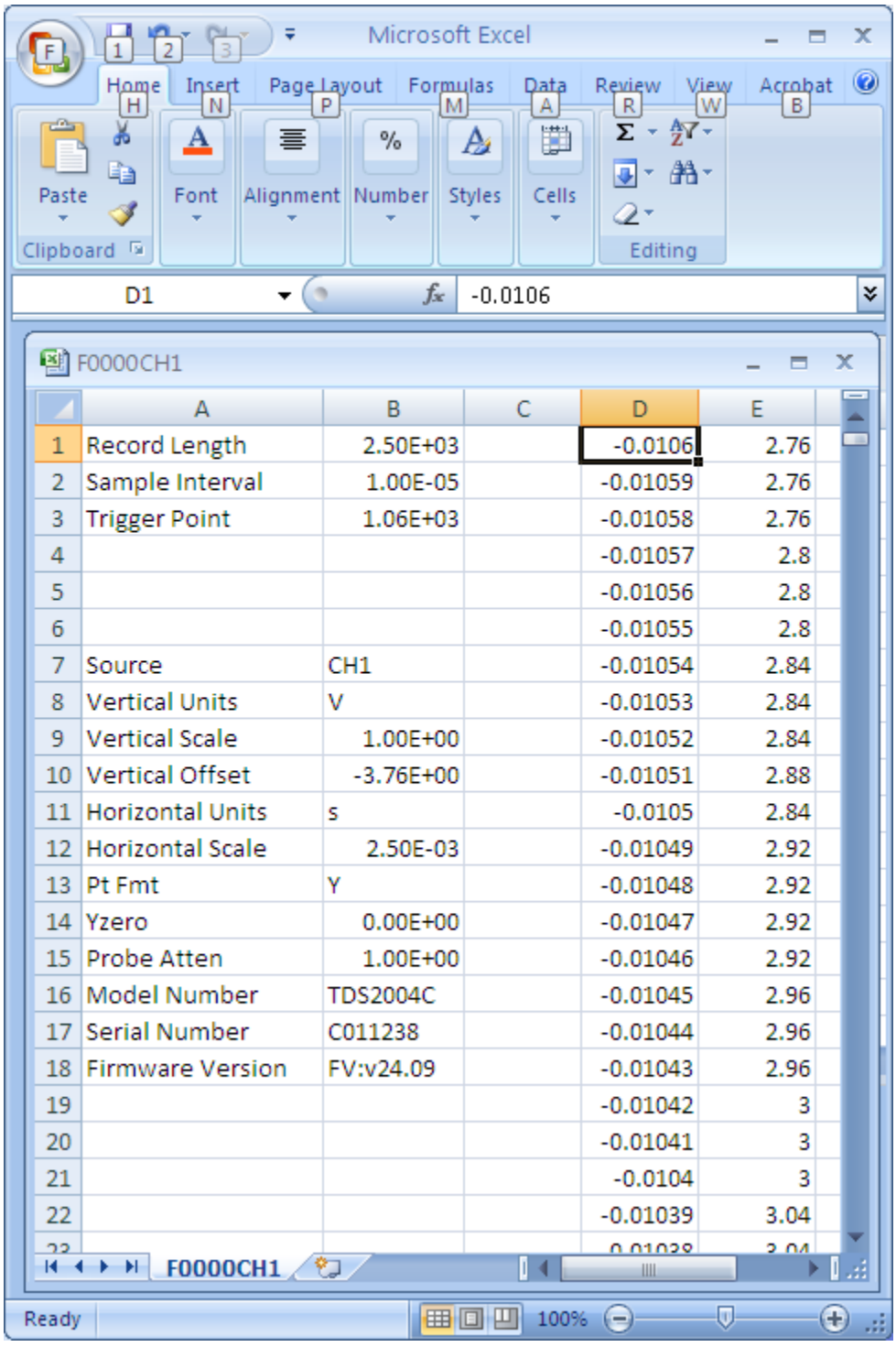
In this case, F0000CH1 and F0000CH2 are .csv (comma separated variable) files with the time and voltage data points for channels 1 and 2 respectively . F0000TEK is the jpeg image of the scope screen, and F0000TEK.SET is a file containing setup information of the scope (channel resolution, AC/DC coupling, seconds/division on the time base, etc.).

MATLAB needs to know which files have the data; this is done with the string variables

```
ch_1_file='F0008CH1.csv';
ch_2_file='F0008CH2.csv';
ch_3_file='F0008CH3.csv';
```

in the beginning of the code.

You can open one of the channel .csv files with excel (or any program that can read a text file) and you will see the following:

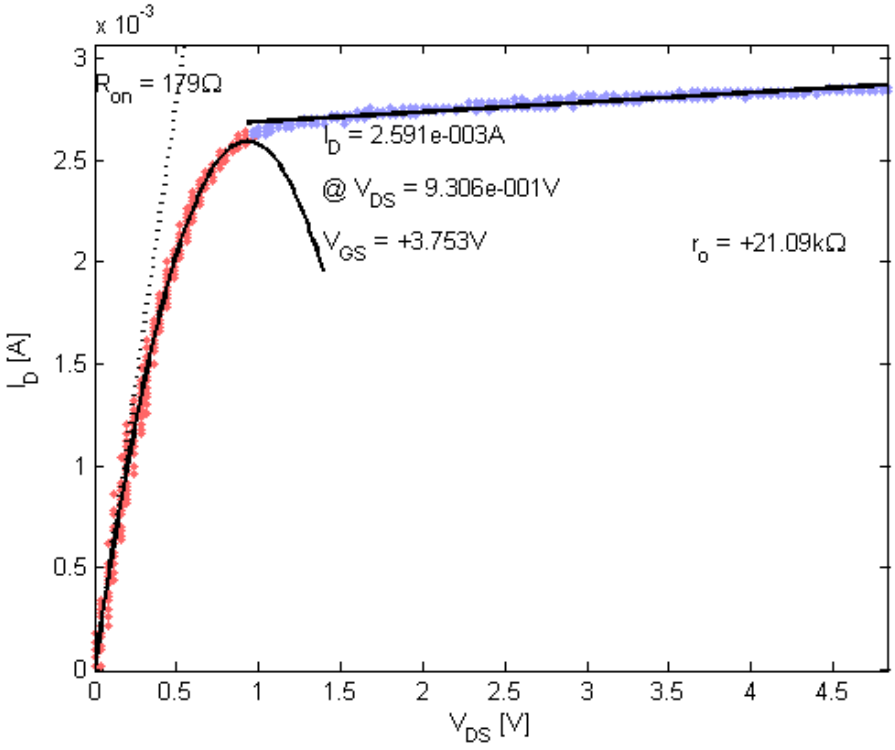


	A	B	C	D	E
1	Record Length	2.50E+03		-0.0106	2.76
2	Sample Interval	1.00E-05		-0.01059	2.76
3	Trigger Point	1.06E+03		-0.01058	2.76
4				-0.01057	2.8
5				-0.01056	2.8
6				-0.01055	2.8
7	Source	CH1		-0.01054	2.84
8	Vertical Units	V		-0.01053	2.84
9	Vertical Scale	1.00E+00		-0.01052	2.84
10	Vertical Offset	-3.76E+00		-0.01051	2.88
11	Horizontal Units	s		-0.0105	2.84
12	Horizontal Scale	2.50E-03		-0.01049	2.92
13	Pt Fmt	Y		-0.01048	2.92
14	Yzero	0.00E+00		-0.01047	2.92
15	Probe Atten	1.00E+00		-0.01046	2.92
16	Model Number	TDS2004C		-0.01045	2.96
17	Serial Number	C011238		-0.01044	2.96
18	Firmware Version	FV:v24.09		-0.01043	2.96
19				-0.01042	3
20				-0.01041	3
21				-0.0104	3
22				-0.01039	3.04

The first two columns contain configuration information. The time and voltage data is in columns 4 and 5. The matlab code is written to import data beginning in column 5, and read in 2500 points (the record length of the TDS2004B scope memory).

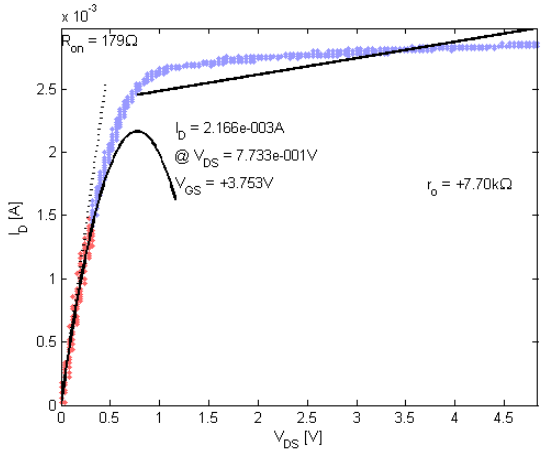
To fit parameters accurately, the MATLAB code needs to know what V_{GS} is. One good way to do that is to use channel 3 of the scope to measure V_{GS}. It's not as accurate as the DVM, but it's pretty good. (An alternative is to go into the MATLAB code and manually enter the V_{GS} you measured using the DVM).

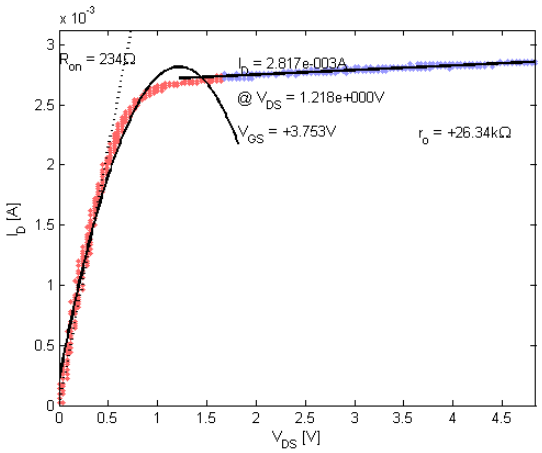
The other thing the MATLAB code needs to know is what to call the triode region for doing the parabolic fit. That is done using the `nfit` variable in the MATLAB code. For now I just have the number chosen and you have to eyeball it from the data. The plot identifies the points chosen for the triode fit by plotting them in red. The plot below shows the output when `nfit` is chosen correctly.



You can tell it's correct because the parabola follows the red triode points pretty closely.

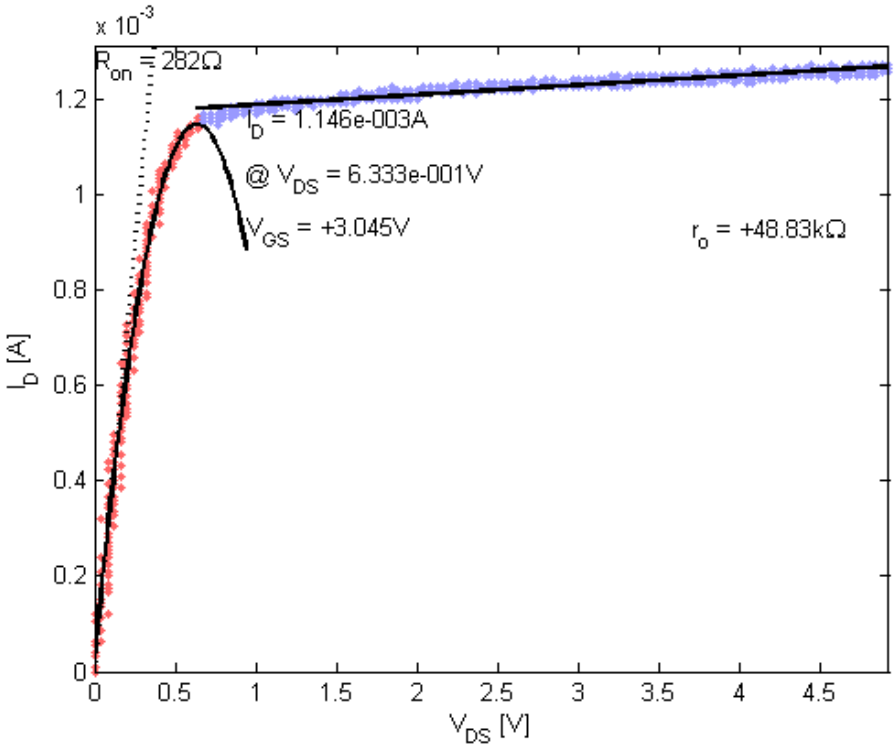
The following plots show the effect of choosing nfit too small or too large.

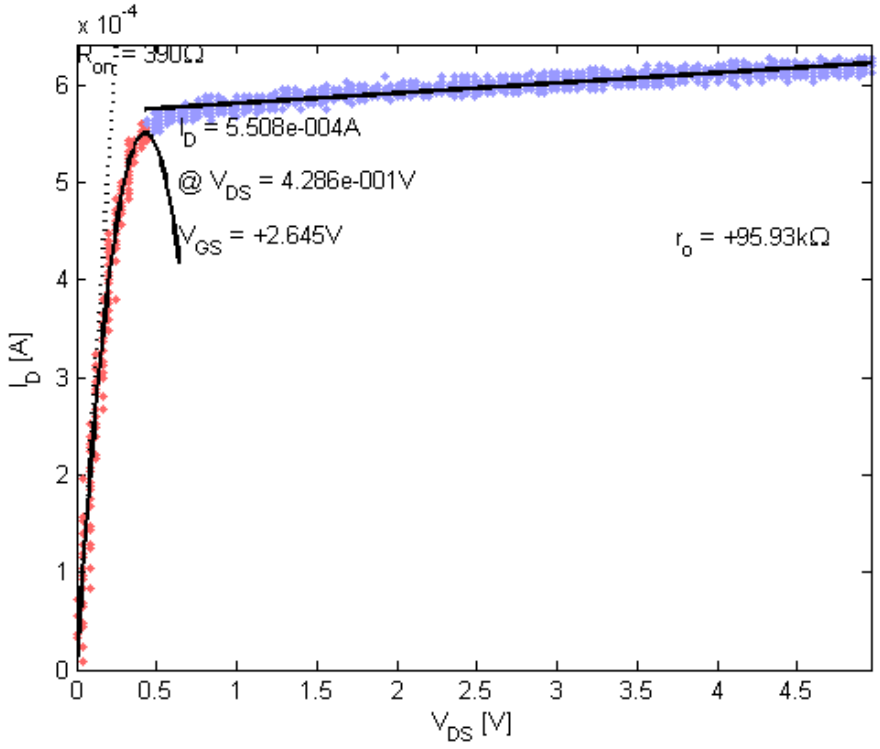


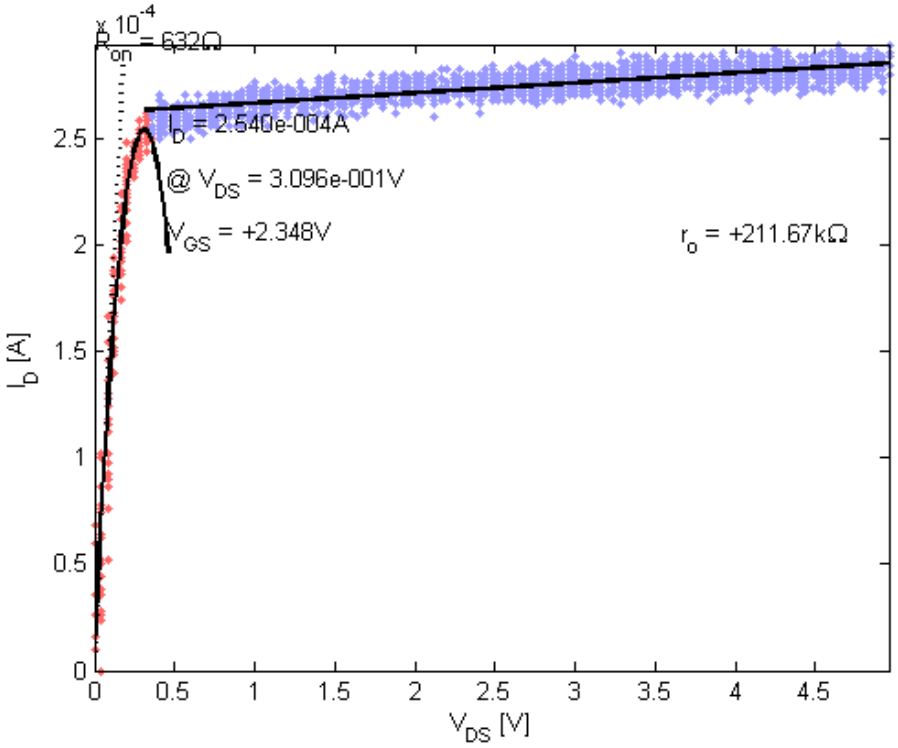


Note that the fit for output resistance is also thrown off when the triode region is incorrectly identified.

If you want, you can repeat the procedure with different values of V_{GS} , capturing several different corresponding files of data, and modifying the matlab code to read in the different files for the .csv data. Each plot gives you the drain current I_D, on resistance R_{on} in the linear part of the triode region, and small signal output resistance r_o in the saturation region.




x 10^-4
R_on = 390Ω
I_D = 5.508e-004A
@ V_DS = 4.286e-001V
V_GS = +2.645V
r_o = +95.93kΩ
I_D [A]
V_DS [V]


x 10^-4
R_on = 632Ω
I_D = 2.540e-004A
@ V_DS = 3.096e-001V
V_GS = +2.348V
r_o = +211.67kΩ
I_D [A]
V_DS [V]