Mathematics 136 – Calculus 2
Lab Day 1 – A Maple "Guided Tour" – Exponential Growth and Moore's Law
September 19, 2016

Background

Our goal today is to start to work with the *Maple* computer algebra system and see some of its capabilities. There is no formal assignment and you will not be handing in anything from today's work. Instead, guided by the instructions below, you will be creating an interactive document for your own reference containing example Maple commands, output, comments, etc. In Maple these are called *worksheets*.

Your Maple worksheet will contain calculations related to a situation some of you may have discussed in other courses–exponential growth and a technological prediction made by the electrical engineer Gordon Moore in 1965. Moore predicted that the number of transistors that could be fit on an integrated circuit chip would *double roughly every two or three years* over an extended period. The fact that he was essentially correct has made possible today's incredibly powerful yet compact electronic devices like "smart phones," tablet and laptop computers, etc. possible(!)

Starting the "guided tour"

Note: As you go through the following "guided tour" you may find you want to enter text comments on what you did and/or saw. By all means, go ahead and do that as desired.

I. Launch Maple and open a worksheet as directed in the "General Information on Maple in Haberlin 136" handout. Enter your name, today's date, the course number (MATH 136) for this course, and the title "Exponential Growth and Moore's Law" in a text region at the top of the worksheet. Suggestion: Use the toolbar button to create an input prompt [>.

II. Enter the following input command at your prompt (don't worry about the line breaks shown here, just keep typing, and Maple will "wrap around" automatically to a new line):

```
ChipData := [[6, 2250.], [7, 2500.], [9, 5000.], [13, 29000.], [17, 120000.],
             [20, 275000.], [24, 1180000.], [28, 3100000.], [32, 7500000.],
             [34, 24000000.], [35, 42000000.], [43, 1900000000.]];
```

This creates a list of data points as shown in problem 29 on page 327 of our textbook, showing a series of points with first component a number of years after 1965 and the second component a number of transistors on the chip. For instance, the pair $[24, 1180000]$ represents the Intel 486 DX processor that came out in 1989 = 24 years after 1965 – the 486 DX processor had $1,180,000$ transistors.

III. At another input prompt, enter the command:

```
plot(ChipData, style = point, color = red);
```

and press ENTER. The output will show a point point of the ChipData dataset. Note that this is not so informative. Why not?

IV. "Doubling every two or three years" should mean an exponential function describes the number of transistors as a function of time. Does an exponential function fit this data well? Visually we can see the answer is pretty clearly "yes" like this: Enter

```
ChipDataLog := [seq([ChipData[i][1], ln(ChipData[i][2])], i = 1 .. nops(ChipData))];
```

What happened here? Well, the `seq` command is *Maple*'s way to generate sequences one term at a time. Here, we're creating a new list of data points where we have applied the *natural logarithm* function to each of the second components of the original `ChipData` dataset. Now enter

```
plot(ChipDataLog, style = point, color = red);
```

What do you notice about this transformed point plot?

V. The *best-fitting* straight line for a dataset is computed via the least-squares regression method that some of you may have encountered in a statistics course or some science course involving data analysis. In *Maple*, we can do the computation very directly like this:

```
with(CurveFitting);
RegLine := LeastSquares(ChipDataLog, t);
```

The *with* command loads a *CurveFitting* package that is not part of the basic Maple system. This sort of thing needs to be done for several other commands we'll see in a moment too. The *LeastSquares* command computes the linear function of the variable `t` specified in the command that gives the least squares regression line.

VI. To see how close this line comes to matching the dataset, let's combine the point plot of the `ChipDataLog` dataset with an ordinary plot of the regression line. The *Maple* commands following load another `plots` package of plotting commands–this contains the `display` command in the following line and must be done for you to have access to this. It is the `display` command that combines plots.

```
with(plots);
Pts := plot(ChipDataLog, style = point, view = [0 .. 50, 0 .. 30], color = blue):
Line := plot(RegLine, t = 0 .. 50, color = red):
display(Pts, Line);
```

The second and third lines here are *assignments* that take the plot command output and associate the name on the left to that data. Then the `display` command takes those plots, combines them, and displays them both on the same axes.

VII. We might also want to see how well an exponential graph seems to match the original data. The following commands do that. Try to figure out for yourself what these do.

```
expform := expand(exp(RegLine));
ExpPlot := plot(expform, t = 0 .. 30, color = red):
FirstThirty := plot([seq(ChipData[i], i = 1 .. 8)], style = point, color = blue):
display(ExpPlot, FirstThirty);
```

Why did I only plot the first thirty years of the exponential function graph and the first 8 points of the `ChipData` dataset?

VIII. There is extensive online Help within the *Maple* system. For instance look up the `plot`, `seq` and `display` commands to see the exact syntax that *Maple* is expecting, various options you can specify, and examples showing how they are used.

IX. If you have extra time, you can try your hand at the following question: Assuming that population growth is nearly exponential, which of the following two sets of data (population in millions of people) is more likely to represent the population of a city over a five-year period? Try to explain your reasoning in a text region.

Year	2000	2001	2002	2003	2004
Set1	3.14	3.36	3.60	3.85	4.11
Set2	3.14	3.24	3.54	4.04	4.74