PHYS 210: Introduction to Computational Physics Fall 2012 Homework 2
Due: Tuesday, October 16, 11:59 PM

PLEASE report all bug reports, comments, gripes etc. to Matt: `choptuik@physics.ubc.ca`

Please make careful note of the following information and instructions:

1. The following assignment requires
   (a) Working with the `xmaple` graphical user interface (GUI) to produce `Maple` worksheets (Problems 1 and 2).
   (b) Preparing source code for `Maple` procedures in plain-text files that can be input into `maple` or `xmaple` via the `read` command (Problems 3, 4).
2. In order to complete your homework—especially for Problems 1 and 2—it is recommended that you use the computer lab. If you try to use `xmaple` remotely, you are apt to find the performance sluggish. However, for Problems 3 and 4, where you are to write procedures in text files, you may be able to work remotely using command line `maple` running on `hyper`.
3. **IMPORTANT!!** To complete the homework, all of the files specified below must exist and be in their proper locations within your `/phys210/$LOGNAME/hw2` directory. The TAs and I must be able to read any and all of the `maple` worksheets and text files into `maple` sessions of our own without encountering errors.
4. Whenever working with *any* worksheet in `xmaple`, be sure to save your work frequently, using, for example, `Ctrl-S`. This will minimize the amount of work that you lose should the interface crash (as it has been known to do from time to time).
5. **IMPORTANT!!** It may take you several *hours* to properly complete Problem 1. You will be provided with lab time to work on it (and the other problems), but you are advised not to leave its completion until the last minute.
6. **IMPORTANT!!** Always start `xmaple` from the command line (i.e. in a terminal), so that the `-cw` option is used: this will ensure that all of the worksheets that you create are in "'classic worksheet" format (this is *required*).
7. Please follow all instructions for each problem carefully, again ensuring that all requested files are in their correct locations—i.e. within subdirectories of `/phys210/$LOGNAME/hw2`—and with the correct names. Also note that any reference to the directory `hw2` below is implicitly a reference to `/phys210/$LOGNAME/hw2`.
8. Do not do any of your work, or save any files, in your home directory or anywhere else that is accessible by your fellow students.
9. Finally, as always, let me know immediately if there is something that you do not understand, or if you encounter serious problems with any part of the assignment.

Problem 1: In your `hw2` directory create a subdirectory `a1`. Using Chapter 2 of the *Maple Learning Guide* that I handed out in class ("Mathematics with Maple: the Basics"), make a facsimile of the `Maple` worksheet I went through in class and save it as `a1.mws` in that subdirectory (i.e. as `hw2/a1/a1.mws`). Note that my worksheet is available online in PDF form via `Course Home Page -> Course Notes -> Maple -> Worksheet [PDF 18 pages] ...` You should refer to both the *Learning Guide*, as well as my version of the worksheet as you complete this problem.

You are to work though Chapter 2 *in its entirety*, essentially entering everything that follows a `Maple` prompt (`>`) into your worksheet. Note, however, that there are examples—such as the use of *side relations* in conjunction with `simplify`—that return an entirely different result in `Maple 16` than in the old version. You should omit these examples, as I did. There are also places where the output from `Maple 16` will be different from those in the Chapter 2 handout, and you should not worry about them—simply type in the expressions as I have done in my worksheet.

Note the following important differences in syntax between the version corresponding to the copied chapter (`Maple V`) and `Maple 16`. My version of the worksheet uses the `Maple 16` syntax and yours should too, of course.

1. The abbreviation for "last result" in `Maple 16` is `%` (percent). In `Maple V` it was `"` (double quote).
2. The concatenation operator in `Maple 16` is `||` (double vertical bar). in `Maple V` it was `.` (dot).

Your worksheet should include annotations corresponding to the various sections and sub-sections of the Chapter, as mine does. Note that I did the annotations in a very simple-minded fashion, by inserting text blocks into the worksheet. This can be done by first selecting the execution group (any part of the worksheet delimited by a left square bracket, `[`) where you want to insert text: select the execution group by single clicking with the left mouse button on the bracket. Then, from the menu bar at the top, choose `Insert -> Paragraph -> Before` or `Insert -> Paragraph -> After`, depending on where you want the text to go. The cursor will then be positioned within this new paragraph (and outside any execution groups), and you can type in the annotation. You can also control the appearance of the annotation by highlighting the text (hold down the left mouse button and sweep over the text), then selecting the font, font size and style (bold, italic etc.) from the pull down menus and buttons that appear whenever you are entering or manipulating text. However, you don't have to worry about making the annotations very fancy.

Finally, please observe the cautions made previously concerning:

1. The time it may take to complete this problem.
2. The wisdom of frequent use of `Ctrl-S`, or some other save mechanism when working with *any* `Maple` worksheet.

Problem 2: Make the subdirectory `hw2/a2`, and within that subdirectory, and using `xmaple`, create a worksheet called `a2.mws` in which the following computations and plotting have been carried out:

$$\frac{\partial^3}{\partial x^2\,\partial y}\left(\left(\cos\left(\frac{\ln(3x+6)}{y}\right)\right)^2\right)\Bigg|_{x=1,y=4} \tag{2.1}$$

$$\int \frac{x^7+6x^3-4}{x^2-1}\,dx \tag{2.2}$$

$$\int_{y=1}^{y=3}\int_{x=1}^{x=2} \frac{x^3-y^2}{x^2+y^2}\,dx\,dy \tag{2.3}$$

Taylor series about $x=0,$ up to and including the $O(x^{10})$ term, of $\sqrt{\cos(x)+\sin(x)+\tan(x)}$ (2.4)

A plot of the error in the above expansion (including the $O(x^{10})$ term), for $0 \le x \le 0.01$ (2.5)

Please note the following important points:

1. For (2.1) and (2.3), your answers *must* result in a *floating point nnumber*, not a general algebraic expression.
2. Your answer to (2.4) *must* include the explicit form of the $O(x^{10})$ term, i.e. the leading-order *error* term should be $O(x^{11})$.

You may find (2.5) a bit more challenging that the other parts of this problem. Define "error" as "exact value - approximate value", and be sure to set `Digits` to a value sufficiently large to produce an accurate plot. Finally, note that you can't plot a Taylor series directly (due to the $O(x^p)$ term that generically appears, and which has no specific value). However, as discussed in class (in the handout "Some Useful Maple Commands", also available via the Maple notes page), you *can* convert a series into a polynomial using the `convert` command. Use

```
> ?convert[polynom]
```

for more details.

Problem 3: Make the subdirectory `hw2/a3` and, within that subdirectory, create a text file `procs` that contains definitions (code) for the following four `Maple` procedures. The procedures must have headers and functionalities as specified:

1. `fcn3 := proc(x::numeric)`
   `fcn3` returns a value defined as follows

$$\begin{array}{rll} x^2+2 & \text{if} & x>0 \\ x^2-2 & \text{if} & x<0 \\ 0 & \text{if} & x=0 \end{array}$$

2. `plot3 := proc(xmin::numeric, xmax::numeric)`
   `plot3` uses the `Maple plot` procedure to generate a plot of `fcn3`, as defined above, on the domain `xmin` $\leq x \leq$ `xmax`.

3. `sumpows := proc(k::integer, n::integer)`
   `sumpows` returns the value of the sum

$$\sum_{i=0}^{n} k^i \tag{1}$$

   with the restriction $n \geq 0$. Your procedure should check for the case $n < 0$ and use the `ERROR` statement to generate an error message as illustrated below. (Note that executing `ERROR` *always* causes a `Maple` procedure to immediately return.)
   Examples:

```
> sumpows(2, 10);
      2047
> sumpows(2, -10);
Error, (in sumpows) n must be >= 0
```

4. `fd := proc(f::procedure, x::float, h::float)`
   `fd` returns an approximation to the first derivative of a `Maple` procedure that defines some function of one variable, $f(x)$. The approximation to $f'(x)$ that is used is

$$f'(x) \approx \frac{f(x+h)-f(x)}{h} \tag{2}$$

   and this is the value that the procedure is to return. Note that in the procedure header `x` is the specific x-value at which the approximation is to be computed.
   Examples:

```
> fd(cos, 0.0, 0.1);
      -0.04995834700
> fd(cos, 0.0, 0.01);
      -0.00499996000
> fd(x -> x^3, 1.0, 0.01);
      3.030100000
> fd(cos, 0.0, -0.1);
Error, (in fd) h must be > 0
```

   Note that the third example uses the $x \rightarrow f(x)$ "arrow notation" for the definition of functions: simply supplying x^3 as the first argument will not work, as you can verify. Finally, note that your procedure should ensure that the supplied value of `h` is > 0. If it isn't, the procedure should use `ERROR` to exit with an error message as shown in the fourth example.

Thoroughly test all of your procedures using input of your own choosing: leave brief comments describing how you did the testing in the file `hw2/a3/README`. Due to the simplicity of these procedures, you do not have to comment them. Again, however, the procedures must be prepared in a *single* `Maple` source file (plain text file) called `hw2/a3/procs`. The TAs and I must be able to read `hw2/a3/procs` into a `maple` or `xmaple` session using the `read` command without encountering errors. We will test your procedures with our own input.

Problem 4: Make the subdirectory `hw2/a4` and, within that subdirectory, create a text file `procs` that contains definitions (code) for the following three `Maple` procedures. The procedures must have headers and functionalities as specified:

1. `lmax := proc(l::list(numeric))`

   `lmax` returns the largest element of the list `l`.

   Examples:

   ```
   > lmax([1,4,2,3]);
         4

   > lmax([1,4.0,2,3]);
         4.0

   > lmax([]);
   Error, (in lmax) null list input is invalid
   ```

2. `lreverse := proc(l::list)`

   `reverse` returns a list in which the elements of its list argument `l` appear in reverse order.

   Examples:

   ```
   > lreverse( [1, 2, 3, 4] );
         [4, 3, 2, 1]

   > lreverse( [ [a, b], c, {d,e,f}, 4, [g, h, i] ] );
         [[g, h, i], 4, {d, e, f}, c, [a, b]]

   > lreverse( [] );
         []
   ```

3. `lnoccur := proc(l::list, val::algebraic, n::integer)`

   `lnoccur` returns `true` if and only if the list `l` contains precisely `n` occurrences of the expression `val`; otherwise it returns `false`.

   Examples:

   ```
   > lnoccur([10,10,20,10], 10, 3);
         true
   > lnoccur([10,10,20,10], 10, 2);
         false
   > lnoccur([foo,bar,foo,bar,foo,foo], foo, 4);
         true
   > lnoccur([foo,bar,foo,bar,foo,foo], dog, 0);
         true
   > lnoccur([], foo, 0);
         true
   > lnoccur([], foo, 1);
         false
   > lnoccur([foo,bar], foo, -1);
         false
   ```

   Note that this procedure *never* outputs an error message (via `ERROR`). In particular, if the input list is null, then it returns `true` if $n = 0$ and `false` otherwise. Also, if $n < 0$ then the procedure always returns `false`.

Test your procedures thoroughly with various input—invalid as well as valid—including null lists (`[]`). Note that the *only* place that a null list should be treated as an invalid argument is in `lmax`. All three procedure definitions should be commented (the commenting doesn't have to be extensive), and, again, must be prepared in a *single* `Maple` source file (plain text file) called `hw2/a4/procs`. The TAs and I must be able to read `hw2/a4/procs` into a `maple`

or `xmaple` session using the `read` command without encountering errors. We will test your procedures with our own input.