PHYS 410/555: Computational Physics Fall 2001 Homework 2
DUE: Tuesday, October 2, 10:00 Report bugs to choptuik@physics.ubc.ca

Important: *The following assignment requires*

1. *Working with the* `xmaple` *graphical user interface (GUI) to produce* Maple *worksheets (Problems 1 and 2).*
2. *Preparing source code for* Maple *procedures in plain-text files which can be input into* `maple` *or* `xmaple` *via the* `read` *command (Problems 3, 4 and 5).*

To complete problems 1 and 2 (i.e. those requiring the GUI), I suggest you use `xmaple` *either via the console of one of the* `lnx` *machines, or on any of the PCs running* `Windows NT` *in Henn 205 or Henn 312.*
If you choose to use the `NT` *PCs, please note the following:*

1. *The* `NT` *machines are reserved for certain periods during the day; see the on-line schedule at*

   `http://www.physics.ubc.ca/clab/labsched.html`

   for more information. Sometime during the week of September 24-28, additional `NT` *PCs should be installed in Henn 205. At this point, there should be (potentially) free seats available even when Physics 120 sessions are being held in the room.*
2. *As mentioned in class, you can login into any of the* `NT` *PCs using your* `physics.ubc.ca` *account name and password, and your own* `physics.ubc.ca` *home directory is then available to you via the* `H:` *drive. You should save all of your worksheets on the* `H:` *drive (i.e. in your home directory on* `physics.ubc.ca`*, or in some sub-directory therein), and, finally,* `scp` *or* `ftp` *them from* `physics` *to the proper location on your account on the* `lnx` *machines. Further instructions are given below.*
3. *When creating a new worksheet which is supposed to have a specific name, such as* `a1.mws` *for Problem 1, I recommend that immediately after opening the new worksheet you select* `Save As ...` *from the* `File` *menu, and then use the* `Save As ...` *window to save the file as (in this example)* `a1.mws` *in drive* `H:`*. Subsequent saves to* `H:a1.mws` *may then be effected via* `Ctrl-S`*. Contact me immediately (or ask someone else for help) if you have any problems doing this.*

Whenever working on **ANY** worksheet in `xmaple`, be sure to save your work frequently, using, for example, `Ctrl-S`.

You can *run* `Xmaple` *on the X-terms (i.e. via* `physics.ubc.ca`*), but it is NOT recommended due to color-map (display) problems.*

Warning: *It may take you several* hours *to properly complete* **Problem 1**—*it is not advised that you leave its completion until the last minute.*

Problems 3, 4 and 5 do not *require the use of the worksheet interface (GUI), and you should thus be able to complete them using "command-line"* `maple`*, from any shell running on the* `lnx` *machines.*

Please follow all instructions below carefully, and ensure that all requested files are in their correct locations **within your `lnx` account** *(with their correct names!) when you have completed the assignment.*

Finally, as always, let me know immediately if there is something which you do not understand, or if you encounter serious problems with any part of the assignment.

Problem 1: Using Chapter 2 of the *Maple V Learning Guide*, make and save a facsimile of the *Maple* worksheet I went through in class. Note that PS and PDF (Portable Document Format) versions of my worksheet are available on-line via the *Class Notes* web page—please refer to those documents as well as the *Learning Guide* itself while doing this exercise. You are to work though Chapter 2 *in its entirety*, essentially entering everything which follows a *Maple* prompt (`>`) into your worksheet. Note, however, that there are several examples which do not work as documented in the *Learning Guide*. You should omit these examples, as I did. Your worksheet should include annotations corresponding to the various sections and sub-sections of the Chapter, as mine does. Observe that complete instructions for adding comments, headings, titles, etc. are available via *Maple*'s on-line help facility. (For example, in `Maple 6`, bring up the main help window—by selecting `Introduction` from the `Help` menu—click on the `Worksheet Interface` hyperlink, then `Overview of Document Processing`, then `Insert Elements into a Worksheet`, etc.; in `Maple V.5`, bring up the main help window—by selecting `Introduction` from the `Help` menu—select `Worksheet Interface`, then `Documenting Your Work`, then `comments`, etc.) When you have finished, ensure that your worksheet, or a copy thereof, is saved as `~/hw2/a1/a1.mws` on your `lnx` account. (Note that `.mws` is the standard extension for `xmaple` worksheet files.) Also observe the cautions above concerning (a) the time it may take to complete this problem, and (b) the frequent use of `Ctrl-S`, or some other save mechanism.

Problem 2: Create a worksheet called `a2.mws` in which the following computations and plotting have been carried out:

$$\frac{\partial^3}{\partial x\,\partial y^2}\left(\sin(xy)\exp\left(\tanh^{-1}(y/x)\right)\right)\Big|_{x=2,y=6} \tag{2.1}$$

$$\int \frac{x^5-6x^3+4}{x^2+6}\,dx \tag{2.2}$$

$$\int_{y=-3}^{y=3}\int_{x=-1}^{x=1}\frac{x^3+4xy+y^2}{x^2+y^2}\,dx\,dy \tag{2.3}$$

$$\text{Taylor series about } x=0, \text{ up to and including the } O(x^{10}) \text{ term, of } \sqrt{\cos(x)+\sin(3x)} \tag{2.4}$$

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

To complete the problem, ensure that your worksheet, or a copy thereof, is saved as `~/hw2/a2/a2.mws` on your `lnx` account. Since there has been confusion about this in the past, please note that your answer to (2.4) problem *must* include the explicit form of the $O(x^{10})$ term. For (2.5), define "error" as "exact value - approximate value", and be sure to set `Digits` to a value sufficiently large to produce an accurate plot.

Problem 3: Write *Maple* procedures as follows:

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

   `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. `lpair := proc(l1::list, l2::list)` ⋯

   If `l1` and `l2` are both lists of length `N`, `lpair` returns a new list, also of length `N`, whose `i`-th element is the 2-element list `[ l1[1] , l2[i] ]`.

   Examples:

   ```
   > lpair([w,x,y,z],[1,2,3,4]);
           [[w, 1], [x, 2], [y, 3], [z, 4]]

   > lpair([],[]);
           [];

   > lpair([1,2,3],[a,b]);
   Error, (in lpair) input lists are not of equal length
   ```

3. `lreduce := proc(l1::list,l2::list(binarynumeric))` ⋯

   The input parameters `l1` and `l2` must be non-null lists of equal length. Further, each element of `l2` must be of type `binarynumeric`, that is, either `0` or `1`. You must make the datatype `binarynumeric` known to `type` by defining the procedure `` `type/binarynumeric` ``:

   ```
   `type/binarynumeric` := proc <your-definition-here> end;
   ```

   Once `` `type/binarynumeric` `` has been appropriately defined, `type` should work as follows:

   ```
   > type(0,binarynumeric);
          true

   > type(1,binarynumeric);
          true

   > type(2,binarynumeric);
          false

   > type(a,binarynumeric);
          false
   ```

   Given that the above constraints are satisfied, the output of `lreduce` is a list consisting of those elements of `l1` which correspond to elements of `l2` which are equal to `1` (with the order of elements of `l1` preserved in the output list).

   Examples:

   ```
   > lreduce([1,4,2,3],[0,1,0,1]);
            [4, 3]

   > lreduce([1,4,2,3],[0,0,0,1]);
            [3]

   > lreduce([1,4,2,3],[0,0,0,0]);
            []

   > lreduce([1,4,2,3],[0,0,1]);
   Error, (in lreduce) input lists are not of equal length

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

```
> lreduce([1,4,2,3],[0,2,0,1]);
Error, lreduce expects its 2nd argument, l2, to be of type
   list(binarynumeric), but received [0, 2, 0, 1]
```

Ensure that your procedures are suitably "bullet-proof"; test them thoroughly with various input—invalid as well as valid—including null lists (`[]`). Have your routines output error messages via `ERROR` when appropriate, as shown in the above usage examples.

All three procedure definitions should be adequately commented, and must be prepared in a *single Maple* source file (plain text file) called `~/hw2/a3/procs`. I must be able to read `~/hw2/a3/procs` into a `maple` or `xmaple` session using the `read` command. Your procedures will be tested with my own input.

Problem 4: Implement a *Maple* procedure that computes the unique polynomial (the Lagrange interpolating polynomial) of degree $n-1$ which passes through $[x_i, f(x_i)],\ i = 1\cdots n$. Note that all of the x_i are assumed to be distinct. The procedure should have the header

```
polyinterp := proc(ldata::list(list),var::name) ...
```

`polyinterp` must return a polynomial in `var`; do not assume, for example, that `var` will always be "x". A sample invocation of `polyinterp` and the resulting output is:

```
> polyinterp([ [0,1], [1,6], [2,4], [3,0] ],'x');

          3       2
  5/6 x  - 6 x  + 61/6 x + 1
```

Prepare the procedure definition (adequately documented and with as much error-checking as possible) in the *Maple* source file (plain-text file) `~/hw2/a4/polyinterp`. Your routine will be tested with my own input. Note that I wrote (will write) this procedure in class; you are free to copy what I did there verbatim. However, you are encouraged to implement the procedure on your own, working from the basic mathematical description, also covered in class. Finally, my version does not exit with an error message if the x_i are not distinct; *yours MUST do so.*

Problem 5: (for 555 credit, ***optional*** **for 410 students)** From the following 3 dimension-full physical constants (values given in *SI* units):

- *Newton's gravitational constant:* $G = 6.673 \times 10^{-11}\,\mathrm{kg}^{-1}\mathrm{m}^3\mathrm{s}^{-2}$
- *Speed of light:* $c = 2.998 \times 10^8\,\mathrm{m\,s}^{-1}$
- *Planck's reduced constant:* $\hbar = 1.0546 \times 10^{-34}\,\mathrm{kg\,m}^2\,\mathrm{s}^{-1}$

it is possible to compute a fundamental mass, length, time, density etc. known as the Planck mass, Planck length, Planck time, Planck density etc. More precisely, for any physical attribute with dimension

$$M^{\alpha_1}\, L^{\alpha_2}\, T^{\alpha_3} \tag{5.1}$$

where M, L, and T have the dimensions of mass, length, and time respectively, and the α_i are real constants, the associated Planck quantity has the same dimensions, and is generically given by

$$c^{\beta_1}\, \hbar^{\beta_2}\, G^{\beta_3} \tag{5.2}$$

for some to-be-determined real constants β_i. For example:

$$L \sim c^{-3/2}\, \hbar^{1/2}\, G^{1/2}$$

where the $\sim$ denotes "has the same dimensions". In *SI* units, then, the Planck length is 1.616×10^{-34}m.

Write a *Maple* procedure called `planck` which accepts algebraic expressions of the form (5.1) and returns the corresponding Planck quantity (5.2). You should first extend the `type` procedure to recognize a new type `MLTdim` which is any expression precisely of the form (5.1) (with *constant* α_i). Thus, for example,

```
> type(M,MLTdim);

            true

> type(M^2/(L*T),MLTdim);

            true

> type(2*L,MLTdim);

            false

> type(M^p1,MLTdim);

            false
```

Once you have extended `type` appropriately, note that you can use *Maple*'s type-checking facility by using a header of the form:

```
planck := proc(x::MLTdim)
```

Finally, extend the floating-point evaluation routine, `evalf` so that it recognizes the constants `G`, `c` and `hbar` and returns their *SI* values (without dimensions) as given above. Prepare all of the procedures you write in a single file called `~/hw2/a5/planck`. Typical output from `planck` should look like this:

```
> # Input definitions from file `planck`
> read planck;

> planck(L);
                           1/2  1/2
                       hbar    G
                       ------------
                            3/2
                           c
> evalf(%);
                                -34
                .1616058223*10
```

Test your implementation thoroughly; it will be evaluated using input of my own design.