PHYS3071/7073 - Due Friday 28th March, 2014 by 11am

Assignment 3 —- Legendre polynomials

Introduction: Legendre polynomials are solutions to the differential equation

$$\frac{d}{dx}\left[(1-x^2)\frac{d}{dx}P_n(x)\right] + n(n+1)P_n(x) = 0 \tag{1}$$

They are frequently encountered in physics and other technical fields, for example when calculating potentials by means of the multipole expansion. An analytic solution for the $P_n(x)$ exists and is given by:

$$P_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}\left(x^2-1\right)^n \tag{2}$$

However for practical purposes it is much more convenient to calculate Legendre polynomials via a recursion relation. The value of the first two Legendre polynomials at a point x is given by:

$$P_0(x) = 1, \quad P_1(x) = x \tag{3}$$

Values of subsequent polynomials at a point x can then be calculated through the following recursive formula:

$$nP_n(x) = (2n-1)xP_{n-1}(x) - (n-1)P_{n-2}(x) \tag{4}$$

Problem 1/6: Write a function that takes as argument any value of x and any positive integer value n and calculates the value of the corresponding Legendre polynomial $P_n(x)$. $P_n(x)$ should be calculated using the above recursive formula.

Problem 2/6: Write a program called `as03-surname-studentid-1.c` that reads from standard input a floating point number x and an integer n, and then prints out $P_n(x)$ using the function that you wrote in Problem 1.

Problem 3/6: Run the program multiple times for the following values of (x, n):

x	0.0	0.5	0.5	-0.5	1.0	-1.0	10.0
n	1	1	5	10	5	10	10

Use the terminal to highlight, copy, and paste the standard output from these runs into a text file with the name `as03-surname-studentid-output.dat` (all lowercase), and then insert a line at the beginning of the text file with `COURSENUMBER AS03 PROBLEM 3 SURNAME STUDENTID` (case doesn't matter).

Problem 4/6: To make your program a little more interesting, modify it such that it only asks the user for the integer number n and then calculates 100 different values of x equally spaced between $x = -1$ to $x = 1$. The program should print the x and $P_n(x)$ values into a file with one pair of values per line.

Problem 5/6: Run this program for the following values of n: $n = 2, 3, 5$ and move the output into different files using the Linux `mv` command (or let the program ask the user for the filename and use

different names for the files). Then use `gnuplot` to show the curves for the different n in one plot. Name this plot `as03-surname-studentid-plot.ps`. Make sure that curves are properly labeled and use as title for the plot `AS03 SURNAME STUDENTID`.

Problem 6/6 (Advanced): The associated Legendre polynomials $P_l^m(x)$ can be defined through the following recursion relation:

$$\sqrt{1-x^2}P_l^{m+1}(x) = (l-m)xP_l^m(x) - (l+m)P_{l-1}^m(x) \tag{5}$$

The above relation holds for all integer m with $0 \leq m \leq l$ and for any integer l and for $-1 < x < 1$. For $m = 0$ the associate Legendre polynomials reduce to the corresponding Legendre polynomials with $l = n$, i.e. $P_l^0(x) = P_n(x)$. Modify your program from problem 2 such that it asks the user for m, l and x and calculates the value of the associated Legendre polynomials $P_l^m(x)$ through the two recursion relations given by eqs. 4 and 5. Your program should stop for any invalid input with an error message. Copy your program to a file `as03-surname-studentid-2.c`

The full set of files you will have when you are done should be:

- `as03-surname-studentid-1.c`
- `as03-surname-studentid-2.c` (for part 6)
- `as03-surname-studentid-plot.ps`
- `as03-surname-studentid-output.dat`

As always, to submit your assignment, mount your directory on the submission server, create a subdirectory called `as03` under your student ID, eg. `s1234567/as03/` and copy your files into this subdirectory. You must submit your files into this subdirectory. Be sure to follow the above submission instructions, otherwise you make the marking of the assignments unnecessary complicated for us.

PHYS3071/7073 - Due Friday 28th March, 2014 by 11am

Grading Sheet – Assignment 3: Legendre polynomials

A: /40% **Function**: Does the program run and produce the correct output?

B: /10% **Usability**: Is the program easy to use? Are the input requirements and output formatting easy to understand?

C: /10% **Readability**: Is the program easy to read and comprehend? Is it well-commented? If the code is sufficiently complex, has it been broken up into manageable subroutines, each of which is well-documented?

D: /10% **Efficiency**: Does the program run efficiently? Is the coding clunky or unnecessarily complicated?

E: /10% **Presentation (Plots)**: Does the plot clearly convey the results? Does it have an appropriate title? Are the axes and the plot items clearly labeled? Was the correct style (points or lines) used for each item?

F: /20% **Advanced part**: Grading for this part follows the same rules as detailed under points A to D.

Total Points: | /100 |