

For the following two problems, write and debug MATLAB codes and make sure they run with the test autograders from the course web page. Test them thoroughly on test cases of your own design. When you are convinced they work, submit your codes together with

- brief discussion of any design decisions
- brief comparison of test results with theory

**Part 1** Implement two MATLAB functions `cleg.m` and `pleg.m` of the form

```
function c = cleg(n)
% n: number of recurrence parameters

function [ p, pp, ppp ] = pleg(x, c)
% x: vector of evaluation points
% c: n-vector of recurrence parameters
```

These functions work together to evaluate all Legendre polynomials  $P_n$  of degree up to and including  $n$ , at an array of evaluation points  $x_j$  with  $|x_j| \leq 1$ . Here  $P_0 = 1$ ,  $P_1(x) = x$  and  $P_n$  is determined by the recurrence

$$P_n(x) = xP_{n-1}(x) - c_n P_{n-2}(x)$$

for  $n \geq 2$ . (As a consequence,

$$P'_n(x) = P_{n-1}(x) + xP'_{n-1}(x) - c_n P'_{n-2}(x)$$

and similarly for  $P''_n(x)$ .) Each parameter  $c_n$  is determined by

$$c_n = \frac{\int_{-1}^1 x P_{n-1}(x) P_{n-2}(x) dx}{\int_{-1}^1 P_{n-2}(x)^2 dx}$$

for  $n \geq 2$ , so for example  $c_2 = 1/3$ . Your code `cleg` should evaluate the integrals to 12-digit accuracy with your code `gadap.m` from Problem Set 07. After `cleg( n )` generates the constants  $c_1, c_2, \dots, c_n$ , your code `pleg` should evaluate  $P_n(x_j), P'_n(x_j), P''_n(x_j)$  by recurrence. `pleg` determines  $n$  by checking the length of the parameter vector  $c$ .

**Part 2** Implement a MATLAB function `gaussint.m` of the form

```
function [w, x] = gaussint( n )
% n: Number of Gauss weights and points
```

which computes weights  $w$  and points  $x$  for the  $n$ -point Gaussian integration rule

$$\int_{-1}^1 f(x) dx \approx \sum_{j=1}^n w_j f(x_j).$$

(a) Find the points  $x_j$  by your code `schroderbisection` from Programming Project 1, modified as necessary. Bracket each  $x_j$  initially by the observation that the zeroes of  $P_{n-1}$  separate the zeroes of  $P_n$  for every  $n$ . Thus the single zero of  $P_1 = x$  separates the interval  $[-1, 1]$  into two intervals, each containing exactly one zero of  $P_2$ . The two zeroes of  $P_2$  separate the interval  $[-1, 1]$  into three intervals, and so forth. Thus you will find all the zeroes of  $P_1, P_2, \dots, P_{n-1}$  in the process of finding all the zeroes of  $P_n$ .

(b) Find the weights  $w_j$  by applying `gadap.m` to

$$w_j = \int_{-1}^1 L_j(x)^2 dx$$

where  $L_j$  is the  $j$ th Lagrange basis polynomial for interpolating at  $x_1, x_2, \dots, x_n$ .

**Code Submission:** Upload the MATLAB files `cleg.m`, `pleg.m`, and `gaussint.m` and any supporting files to bCourses for your GSI to grade.