CSCI-GA.2420-001/MATH-GA.2010-001
Numerical Methods I Fall 2013
October 29, 2013

Professor Olof Widlund
Office: CIWW 612, Phone: 998-3110
Electronic mail: widlund@cims.nyu.edu

Homework set 5: Due Friday November 8 at 12 noon.

Homework should be given to me in class or put under my office door. Do not put it in my mail box.

In all cases, discuss your findings.

1. Recall that $\mathcal{P}_n$, the set of all polynomials of degree n or less, is a classical example of a finite dimensional vector space if vector addition and multiplication by scalars are defined in the obvious way.

   (a) Show that operator $A = -\frac{d}{dx}((1-x^2)\frac{d}{dx}$ maps $\mathcal{P}_n$ into itself for any n. Also show that this operator is satisfies

   $$\int_{-1}^{1}(Ap(x))q(x)dx = \int_{-1}^{1}p(x)(Aq(x))dx$$

   .

   (b) Show that the operator A_n, A restricted to $\mathcal{P}_n$, has a full set of eigenvectors with $n+1$ real eigenvalues, and that A_{n-1} and A_n share n eigenvalues.

   (c) Show that the Legendre polynomials $L_n(x)$ satisfy

   $$\frac{d}{dx}((1-x^2)\frac{d}{dx}L_n(x) + n(n+1)L_n(x) = 0.$$

   (d) Show that $L'_n, n \geq 1$, the first derivatives of the Legendre polynomials, form a set of orthogonal polynomials with respect to the weight $1-x^2$ and the interval $(-1,+1)$.

   (e) Consider the quadrature rule based on the zeros of $(1-x^2)\frac{d}{dx}L_n(x)$. What is its order, i.e., what is the largest set polynomials which is integrate exactly by this *Gauss–Lobatto* rule.

2. Consider the case when $x = 1$ is a quadrature node while all other nodes are chosen to maximize the order of the quadrature rule. How are the nodes and quadrature weights obtained? What is the order of this quadrature rule?

3. Write and test a program for the adaptive Simpson quadrature method. (This topic is covered on pp. 328–331 in an October 24 handout.) The integrand should be defined by a matlab function which provides values of the integrand $f(x)$ for any given input value x.

   The adaptive Simpson quadrature algorithm is recursive and if your program is properly designed the value of $f(x)$ should never be computed more than once for any particular value of x. You can check if you do it right by counting the number of quadrature nodes and the number of function calls.

   When the program is running, there is an active interval the contribution of which to the final approximate value of the integral, we are trying to compute accurately enough. The overall tolerance, a positive number, ϵ, is provided as input. The contribution of each subinterval, to the overall error should not exceed $\epsilon*$ the length of the subinterval measured as a fraction of the entire, given interval.

   The first active interval is the entire interval. If we decide that the quadrature rule is not accurate enough, the left half of the active interval becomes the active interval. Once the contribution from this interval has been computed accurately enough, we compute the contribution of the right half of the interval. This is a recursive procedure.

   You can either use recursive calls or construct *stacks* to accomplish the savings of the function values, already computed, that you need later. A stack can easily be implemented using a vector and an index that serves as a pointer.

4. Test your program by finding the approximate value, for several values of the tolerance, of

$$\int_0^1 x^{1/2} dx,$$

$$\int_0^1 (1 - x^2)^{3/2} dx$$

   and

$$\int_0^1 \frac{\sin(x)}{x^{3/2}} dx$$

and compare the results with the exact values of the integrals if they are available.

Note that the third integrand takes on an infinite value at one end point. Find an appropriate device to deal with this so as to obtain accurate values of the integral.

5. Find similar algorithms available in matlab and evaluate the same integrals. Compare the results.

6. Write and test a program for Clenshaw-Curtis quadrature and test them on some examples such as those given above. (This quadrature rule will be discussed in some detail in the October 31 lecture.) Discuss the relative merits of this quadrature rule and adaptive Simpson.