Assignment Set 4, MATH-UA0252-1

Olof Widlund

March 23, 2015

The following assignments are due on Monday April 6 at 12:00 noon. No homework will be accepted after that time. You should give me your homework in class or put it under my office door (WWH612). Do not use my mailbox in the WWH lobby.

You should turn in your MATLAB/OCTAVE programs and printouts which shows how they work in different cases. In addition, is is important that you comment on what you can learn about the performance of the methods. Always use format long e in MATLAB. Note that in MATLAB, all numbers are represented with approximately 16 decimals; we will discuss computer arithmetic in the middle of the term.

Handouts that are relevant to these problems will be given out on March 24. The first two problems relate to polynomial interpolation on the interval $[-1, +1]$ interpolating at the Chebyshev points $\cos(j\pi/n),\ j = 0, \ldots, n$. Instead of the standard monomial basis for the space of polynomials, we can use the Chebyshev polynomials $T_k(x),\ k = 0, \ldots, n$.

1. Show that the coefficients of the Chebyshev series which interpolates given values $f_i,\ i = 0, \ldots, n$, can be computed by using FFT, in particular a cosine transform. Write a program that implements this algorithm. Note that the Chebyshev series expansion will have rapidly decaying coefficients if the function we interpolate is smooth; the same is true of Fourier series expansions of periodic functions.

2. The Clenshaw-Curtis numerical quadrature rule is derived by integrating the Chebyshev series that interpolates a given function at the Chebyshev nodes and using the fact that $\int_{-1}^{+1} T_k(x)dx = \frac{2}{1-k^2}$ for $k = 0, 2, 4, \ldots$ while the integrals of T_k vanishes for all odd k. Use convenient values of n and study the rate of convergence when n successively is doubled for some nice, smooth functions as well as for the rather nasty function

   $$e^x[sech(4\sin(40x))]^{exp(x)}.$$

   Recall that $sech(x) := \cosh(x)^{-1}$.

3. Write and test a program for the adaptive Simpson quadrature method. (This topic is covered on pp. 328–331 in a March 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 should 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 to obtain an accurate value.

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