V63.0252 **Numerical Analysis** **Spring 2007**

Arpil 2, 2007

Professor Olof Widlund
Office: CIWW 712, 251 Mercer Street
Phone: 212 998-3110
Electronic mail: widlund@cims.nyu.edu
Course home page URL: http://www.math.nyu.edu/courses/spring07/V63.0252-001/index.html
Office hours: Mondays 3:30–4:30pm and Thursdays 4:00–5:00pm.

Homework set 6: Due Wednesday April 25, at midnight.

No homework will be accepted after that time.

Homework should be given to me in class or put under my office door. Do not put it in my mail box. For general rules, read the homepage.

When running matlab, use **format long e**.

In all the problems, use plots to demonstrate the distribution of the quadrature nodes.

1. Write and test a program for the adaptive Simpson quadrature method. (This topic is covered on pp. 328–331 in one of the handouts.) 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.

2. Test your program by finding the approximate value, for several small 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 infinite values at one end point. Find an appropriate device to deal with this so as to obtain accurate values of the integral.

3. Find similar algorithms available in the MATLAB system and evaluate the same integrals. Compare the results.

4. Do problems 7.4 and 7.5 in the textbook.