The overall objective is to improve the text’s adaptive quadrature program No. 7.6. We will replace Simpson’s rule by the three point Gauss’ rule.

In order to get the cleanest code we will use recursion. Recall that the adaptive quadrature algorithm is defined recursively. In such a case a recursive implementation is most likely to be simpler and correct.

Of course there is the issue of efficiency. It is often the case that a straightforward coding of a recursive algorithm is seriously inefficient. Such is the case here. The main “cost” in the algorithm is the number of evaluations of the integrand. So your implementation must not evaluate the integrand more than once at each point.

Also, go back and completely understand the derivation of the adaptive quadrature algorithm. It is not enough to merely replace the calls to Simpson’s rule with calls to Gauss’ rule. The decision about when to stop recursing and accept the approximation changes because the error term is different between Simpson’s rule and Gauss’ rule. Part of your assignment is to figure out what the new stopping condition should be.

You should break the programming down into several subprograms. (They can all be put into one file. This is very convenient while debugging. Be sure to end each program therein with “endfunction.”) In mine I have functions for the two different integrands, three point gauss, the main recursive part, and a very short calling program to start the recursion properly. To give an idea of what is possible, none of my programs is more than 5 lines long, although I do group related statements on the same line, oftentimes.

Put in “scaffolding” which writes out the intervals which are not further subdivided and also information which allows the number of calls to evaluate the integrand to be counted. It should be possible to “turn off” the scaffolding by “commenting out” certain obvious lines (marked in advance by a comment). In my line counts above I was not counting the scaffolding lines.

Check your programming against Example 7.16 of the text and say 6 equally spaced values of the function erf(x), \(1 \leq x \leq 2\), 10D on the handout.

Hand in a description of what each program does, how they work together, copies of all programs, computer runs showing the results indicated in the last
paragraph above. Also, include your stopping condition mentioned above and how you arrived at it.

Good luck. Remember I am ready, willing and able to help. It is not an imposition.