Program 4 Foundations of Computational Math 2 Spring 2022

Due date: 11:59PM on April 10, 2022

Programming Exercise

In this assignment you will implement numerical quadrature methods and compare their observed behavior to theoretical predicitons.

The Codes: Consider the composite quadrature methods based on the following basic methods:

- The closed Newton-Cotes method that uses two points (the two endpoints), i.e., the Trapezoidal Rule.
- The closed Newton-Cotes method that uses three points (the two endpoints and the midpoint), i.e., Simpson's First Rule.
- The open Newton-Cotes method that uses one point (the midpoint), i.e., the Midpoint Rule.
- The open Newton-Cotes method that uses two points (the points at $1/3$ and $2/3$ across the interval of integration).
- The two-point Gauss-Legendre method.

1.1. For each of these basic methods implement a composite quadrature method that uses a set of uniform subintervals of the interval of integration $[a, b]$. (Note that you use the basic methods above on each of the subintervals.)

1.2. For each of these basic methods implement a composite quadrature method that uses a set of uniform subintervals of the interval of integration $[a, b]$ and global uniform subinterval size refinement with a factor $\alpha = 1/k$. For the composite Gauss-Legendre method take $\alpha = 1/2$. For the open and closed Newton-Cotes take k to be such that there is complete reuse of the function evaluations on the coarse grid (and all earlier coarse grids), i.e., so you only need to evaluate $f(x)$ at the new points on the fine grid. We know $\alpha = 1/2$ accomplishes this for the composite Trapezoidal rule.

Provide your justification for your choice of α for the other Newton-Cotes methods. You must use the largest $\alpha = 1/k$ that achieves complete reuse for each Newton-Cotes method. **(Your implementations should be as efficient as possible in terms of function evaluations. Note however, as mentioned in class, that for methods with complete reuse the composite method code with refinement can be used as the implementation of the composite method without refinement. For methods with no reuse you may also be able to use the same code but this should be considered carefully. You may, if you prefer, implement to distinct codes for each form of the composite methods. Make sure you explain your approach in your solutions.)**

The Integrals: The integrals to be evaluated include

$$\int_0^3 e^x dx = e^3 - 1 \tag{1}$$

$$\int_0^{\frac{\pi}{3}} e^{\sin(2x)} \cos(2x) dx = \frac{1}{2}\left(-1 + e^{\frac{\sqrt{3}}{2}}\right) \tag{2}$$

$$\int_{-2}^{1} \tanh(x) dx = \ln\left(\frac{\cosh(1)}{\cosh(2)}\right) \tag{3}$$

$$\int_0^{3.5} x\cos(2\pi x) dx = -\frac{1}{2\pi^2} \tag{4}$$

$$\int_{0.1}^{2.5} \left(x + \frac{1}{x}\right) dx = \frac{2.5^2 - 0.1^2}{2} + \ln(2.5/0.1) \tag{5}$$

Note all of these have symbolic solutions that may be used to assess true error, predict expected behavior and analyze observed behavior.

The Tasks:

1.1. Provide systematic evidence of the correct execution of your codes. This will require more problems than those listed above.

1.2. Make sure that you include in the description of your codes sufficient information how your code is designed to exploit the efficient reuse of function evaluations to avoid redundant work based on your derivation of the optimal α required above. For example, how did it change the organization of the computation or data structure when efficient interval refinement is used or not used.

1.3. For each integral in the problems list, estimate the subinterval size needed to satisfy various error demands using the simple composite quadrature methods. The relative error requires knowledge of the size of the answer I. You may use your knowledge of the exact answer for each to make these predictions.

1.4. Run the codes for various error requirements, summarize appropriately and concisely your observations, compare the observed performance to predictions and explain based on knowledge of the methods and the particular problems. Comment on any behavioral differences observed between the problems and explain them based on differences in the functions being integrated. Discuss and compare the number of function evaluations. Your discussions should include comparing the accuracy actually achieved using the true error computed from your analytical solutions to the integrals. Are the subinterval sizes used based on your a priori analysis for the composite methods conservative, i.e., do you get greater accuracy than you expect?

1.5. Repeat the experiments and explanations for the composite codes using your chosen α for each. Compare the results with the methods above that used a subinterval size chosen by a priori error analysis. Specifically, how did the accuracy per function evaluation ratio behave for the two sets of methods.

Other Test Problems

After you have submitted you solutions you should make an appointment with the TA Yue Shen. She will ask you to demonstrate your code on some test problems as a final evaluation of your codes correctness. You should also be prepared to explain the design and operation of your code to her.