MAT 128A, Fall 2016

Programming Project 3

(due before class on Wednesday, November 23)

General Instructions

- The above deadline is hard. I will not accept any submissions after the lecture has started or after class.
- Write a report that includes all required numerical results, a discussion of your results, and printouts of your Matlab programs.
- When you are asked to print out numerical results, print numbers in 16-digit floating-point format. You can use the Matlab command "`format long e`" to switch to that format from Matlab's default. For example, the number 10π would be printed out as `3.141592653589793e+01` in 16-digit floating-point format.
- Upload all your Matlab programs into your drop box on the MAT 128A site on SmartSite by the due date of the project.

We consider the problem of computing approximations to general integrals

$$I = I(f, a, b) = \int_a^b f(x)\,dx,$$

where $a < b$ and $f : [a, b] \mapsto \mathbb{R}$ is an integrable function.

1. Write a Matlab program that implements the version of the composite trapezoidal rule presented in class. Note that this version is designed to use as few function evaluations as possible and that it produces a conservative error estimate E_j along with the final approximation T_j to I.

   The inputs for your program should be:

   - The integration limits a and b;
   - A Matlab function to compute $f(x)$ for any $x \in [a, b]$;
   - An error tolerance `tol`;
   - A safeguard $j_{\max}$ to limit the iteration index j.

   The output for your program should be:

   - The final approximation T_j to I;

- The corresponding conservative error estimate E_j;
- The final iteration index j;
- The total number `nfcount` of function evaluations that your program has used.

2. Write a Matlab program that implements the version of the composite Simpson's rule presented in class. Note that this version is designed to use as few function evaluations as possible and that it produces a conservative error estimate E_j along with the final approximation S_j to I.

   The inputs for your program should be:

   - The integration limits a and b;
   - A Matlab function to compute $f(x)$ for any $x \in [a, b]$;
   - An error tolerance `tol`;
   - A safeguard $j_{\max}$ to limit the iteration index j.

   The output for your program should be:

   - The final approximation S_j to I;
   - The corresponding conservative error estimate E_j;
   - The final iteration index j;
   - The total number `nfcount` of function evaluations that your program has used.

3. Let
$$x_0 := a < x_1 < x_2 < \cdots < x_{n-1} < b =: x_n \tag{1}$$
   be given $n + 1$ points in $[a, b]$, and assume that $n \geq 3$. Recall that there exists a unique not-a-knot cubic spline $s = s(x)$ such that
$$s(x_i) = f(x_i), \quad i = 0, 1, \ldots, n.$$
   We can obtain an approximation to I by integrating $s = s(x)$:
$$I_{\text{spline}} := \int_a^b s(x)\,dx \;\approx\; \int_a^b f(x)\,dx = I.$$
   Write a Matlab program that implements this approach. Recall that Matlab's `spline` function produces a structure that contains all the necessary information about s. In particular, it contains the coefficients of the n pieces of s that you will need to integrate to obtain I_{spline}.

   The inputs for your program should be:

   - A vector of length $n + 1$ containing $x_0, x_1, \ldots, x_n$ from (1);
   - A Matlab function to compute $f(x)$ for any $x \in [a, b]$.

   The output for your program should be:

- The approximation I_{spline} to I.

4. Test each of your three programs for the following integrals:

   - $I = I(b) = \operatorname{erf}(b) := \dfrac{2}{\sqrt{\pi}} \displaystyle\int_0^b \exp(-x^2)\,dx$ for $b = 0.4, 0.8, 1.2, 1.6, 2.0$. Note that you can use Matlab's `erf` function to compute $I(b)$;
   - $I = \pi = 2 \displaystyle\int_{-1}^{1} \frac{1}{1+x^2}\,dx.$

   For all your runs with the composite trapezoidal rule and the composite Simpson's rule, use the tolerance `tol` $= 10^{-12}$ and choose $j_{\max}$ large enough so that $E_j \le$ `tol` is reached before the safeguard $j_{\max}$ forces the algorithm to stop.

   For each of your runs with the composite trapezoidal rule, print out

   $$T_j, \quad E_j, \quad |I - T_j|, \quad j, \quad \text{and} \quad \texttt{nfcount}.$$

   For each of your runs with the composite Simpson's rule, print out

   $$S_j, \quad E_j, \quad |I - S_j|, \quad j, \quad \text{and} \quad \texttt{nfcount}.$$

   For your runs with the spline-based approach, choose $n =$ `nfcount` $- 1$, where `nfcount` is the number of function evaluations from your corresponding run with the composite Simpson's rule. This way, both approaches use exactly the same number of function evaluations. For each integral, run your program with the following two choices of the points (1):

   - Equally-spaced points;
   - Chebyshev points.

   For each of your runs with the spline-based approach, print out

   $$I_{\text{spline}} \quad \text{and} \quad |I - I_{\text{spline}}|.$$

   For all your runs, comment on your results, especially on the efficiency of the three numerical integration methods.