Math 373 Assignment 6 (Due Thursday, October 23.)

1. The trapezoidal rule with end correction is the approximation

$$\int_a^b f(x)\,dx \approx \frac{h}{2}[f(a) + 2f(a+h) + \cdots + 2f(a+[N-1]h) + f(b)] + \frac{h^2}{12}[f'(a) - f'(b)].$$

a) Derive this formula by integrating the piecewise cubic Hermite polynomial interpolating f and f' at x_i, $i = 0, \cdots, N$, where $x_i = a + ih$, $h = (b-a)/N$.

Hint: If you use the Newton form of the interpolating polynomial on each subinterval $[x_{i-1}, x_i]$, then the following formula will greatly simplify the integration of the last term:

$$\begin{aligned}\int_{x_{i-1}}^{x_i} (x - x_{i-1})^2(x - x_i)\,dx &= \int_{x_{i-1}}^{x_i} (x - x_{i-1})^2([x - x_{i-1}] - [x_i - x_{i-1}])\,dx \\ &= \int_{x_{i-1}}^{x_i} (x - x_{i-1})^3\,dx - [x_i - x_{i-1}] \int_{x_{i-1}}^{x_i} (x - x_{i-1})^2\,dx.\end{aligned}$$

b) Assuming $f \in C^4[a,b]$, derive the error formula for this approximation

$$E = \frac{b-a}{720} h^4 f^{(4)}(\xi), \qquad a < \xi < b.$$

2. One form of the error for the approximation of $\int_a^b f(x)\,dx$ by the composite rectangle rule $h\sum_{i=0}^{N-1} f(a+ih)$ (where $h = (b-a)/N$) is given by

$$\int_a^b f(x)\,dx - h\sum_{i=0}^{N-1} f(a+ih) = \frac{h}{2}[f(b) - f(a)] - \frac{b-a}{12} h^2 f''(\mu),$$

for some $a < \mu < b$. By applying Richardson extrapolation to this process, derive another method which is $O(h^2)$ accurate. What is the method?

Before doing Problems 3 and 4, copy the following files from the web page `http://www.math.rutgers.edu/~falk/math373/matlab-prog.html` *to your home directory. The use of these files will greatly simplify these problems.*

- `fcn1.m` This file contains the function `fcn1(x)` which takes as input the value `x` and returns the value `fcn1(x)`.
- `fcn2.m` This file contains the function `fcn2(x)` which takes as input the value `x` and returns the value `fcn2(x)`.
- `mid.m` This file contains the function `mid(FunFcn,a,b,n)` which takes as inputs the name of a function, the left and right endpoints `a` and `b` of the interval of integration, and the number of subintervals `n`, and returns the approximation to the integral of `Funfcn` given by the composite midpoint rule on `n` subintervals.

- `trap.m` This file contains the function `trap(FunFcn,a,b,n)` which is similar to `mid`, but instead returns the approximation to the integral of `Funfcn` given by the composite trapezoidal rule on `n` subintervals.
- `quadtrap.m` Contains function `quadtrap(FunFcn,a,b,tol,ninit,maxn)` taking as inputs the name of a function, the left and right endpoints `a` and `b` of the interval of integration, the absolute error tolerance `tol`, the initial number of subintervals `ninit`, and the maximum number of subintervals allowed `maxn`, and returns a vector whose components are: the approximation to the integral of `Funfcn` given by the composite trapezoidal rule and an interval doubling strategy, the final number of subintervals used, and the error between the last two approximations.
- `quadsimp.m` Contains function `quadsimp(FunFcn,a,b,tol,ninit,maxn)` which has the same inputs and outputs as `quadtrap.m`, but uses the composite Simpson's rule instead of the composite trapezoidal rule.

A typical statement in a *Matlab* program which calls one of these functions is:

```
[v, n, err] = quadtrap('fcn1',0,1,.0001,2,100000)
```

Note that the name of the function must be enclosed in quotes and the output of the function is a vector.

3 (a). Find approximations to the following integrals:

$$\int_0^1 (1-4x(1-x))^{1/3}\,dx \qquad \text{and} \qquad \int_0^1 xe^{-x}\,dx$$

by using the composite trapezoidal rule (`quadtrap.m`) and composite Simpson's rule (`quadsimp.m`). Use the *Matlab* statement `format long` to get extra precision and choose `ninit = 2` and `maxn = 100000`. Run the programs for the choices `tol =` 10^{-2}, 10^{-4}, and 10^{-8} and record the error and the number of subintervals used in a table. In addition, for `tol=` 10^{-16} run only the Simpson's rule program on the second function.

(b) Now compute the order of convergence of these approximations by using the following idea. Assume that the error of each approximation has the form $E = Ch^\alpha$, where C and α are constants. Then if E_1 and E_2 are the errors corresponding to h_1 and h_2, respectively, we have $E_2/E_1 = (h_2/h_1)^\alpha$ from which it follows that $\alpha = \ln(E_2/E_1)/\ln(h_2/h_1)$. Since for this problem, $h = 1/N$, the number of subintervals, $\alpha = \ln(E_2/E_1)/\ln(N_1/N_2)$. For each function and each method, use this formula to compute an approximate α for each pair of *successive* entries in the table of part (a). [In the case of Simpson's rule and the second function, do the computations corresponding to the pairs, $(10^{-4}, 10^{-8})$ and $(10^{-8}, 10^{-16})$, rather than $(10^{-2}, 10^{-4})$ and $(10^{-4}, 10^{-8})$.]

(c) Briefly explain why the results do or do not agree with the order of convergence predicted by the theory developed in class.

4. Write a *Matlab* function whose first line is

```
function [r, nfinal, errfinal] = romb(FunFcn,a,b,tol,ninit,maxn)
```

which may be used to evaluate an integral by Romberg integration, where `r` returns the entire Romberg array, `nfinal` is the largest number of subintervals used, and `errfinal` is the error in the final approximation obtained by the method. Then use this function to evaluate the above two integrals with `tol = ` 10^{-16}, `ninit`$= 2$, and `maxn`$= 100000$. For each integral, hand in the the final approximation obtained by Romberg integration, and the values of `nfinal` and `errfinal`. Explain why the extrapolations do or do not improve the accuracy. You will save a lot of time by making use of the function *m* files provided to you.