Assignment 1.

Given February 4; Parts 1–3 due **February 11**, Parts 4–6 due **February 18**.

Objectives: (1) To warm up to programming with good habits, (2) To illustrate basic numerical ideas, (3) To explore available software.

We want to compute

$$\int_0^1 \cos\left(\alpha x^2\right) dx \quad . \tag{1}$$

for some different values of α.

Step 1: Write a program to estimate the integral using a panel integration method, with the number of panels $n = 1/\Delta x$ as input. The program should be modular, well documented, robust, and clean. This routine should be written so that another person could easily substitute a different panel method by changing a few lines of code. Turn in a listing of this code.

Step 2: Verify the correctness of the program by checking that it gives the right answer for small α. We can estimate the integral for small α using a few terms of its Taylor series. This series can be computed by integrating the Taylor series for $f(\alpha) = \cos(\alpha x^2)$ term by term. Turn in a table showing your Taylor series approximation and the value returned by your code for a few small values of α, say, $\alpha = .1, .2, .3$.

Step 3: With $\alpha = 1$, do a convergence study to verify the second order accuracy of the trapezoidal rule and the fourth order accuracy of Simpson's rule. Turn in a table showing the results for different step sizes Δx, and explain how your results demonstrate the correct order of accuracy.

Step 4: Plot the integrand, $\cos(\alpha x^2)$, for $\alpha = .1,\ 1,\ 10$. Which of these integrals do you think will be the most difficult to compute?

Step 5: Write a routine that uses the trapezoidal rule and Richardson extrapolation to compute the integral to within a specified (absolute) error tolerance. The desired error bound should be input. The output should be the estimated value of the integral and the number of function evaluations used. This routine should be robust enough to quit and report failure if it is unable to achieve the requested accuracy. Turn in a few examples with your input and output from the routine. Try to find an example where your code reports failure.

Step 6: Find a numerical quadrature routine somewhere else. You can use one from *Numerical Recipes* or you can obtain one from Netlib. To see what is available in Netlib, type: netscape http://www.netlib.org . There are also some quadrature routines available in MATLAB. To see what is available in MATLAB, type: help funfun . Find out

what method the routine is using and try the examples from Step 5 with this routine. Report on advantages or disadvantages of this routine over the one you wrote. Which is more robust? Which requires fewer function evaluations to achieve a given level of accuracy?