Name: ____________________ Section: __________

ID#: ____________________

CSE/MATH 455, Arnold | Lab 6 | November 8, 1996

Some Quadrature Rules with Equally Spaced Points

Files to copy: `adaptsimp.m`, `humpsp.m`, and `showadapt.m` from `~dna2/pub/quadrature`.

In this lab we will investigate a simple adaptive quadrature routine, `adaptsimp`, written by me, and Matlab's high-order adaptive quadrature routine `quad8`.

The call for these programs is identical. Call with a string giving the name of the function, the limits of integration, and a tolerance. Thus `quad8('humps',0,3,1.e-10)` computes the integral of the humps function over $[0, 3]$ to high accuracy. One difference between the routines is that for `adaptsimp`, `tol` is an absolute tolerance, while for `quad8`, it is a relative tolerance.

Call `quad8` as above to find the "exact" value of the integral.

Call `adaptsimp` to evaluate this integral with tolerances 10^{-1}, 10^{-2}, ..., 10^{-8}. Check the error and see whether it meets, exceeds, or fails to meet the tolerance.

By putting a counter in the routine for evaluating the function we can count how many function evaluations are used. Putting in plotting statements we can also plot where the function is evaluated. This is done in the function routine `humpsp.m` (humps with plotting). The script `showadapt.m` initializes the counter and graphics window and calls `adaptsimp` to integrate `humpsp`. Run it. You can vary `showadapt.m` in various ways: changing the tolerance, switching from `adaptsimp` to `quad8`.

For various tolerances, (say tol $= 10^{-1}$, 10^{-2}, ..., 10^{-8}), which is more efficient, `adapt-simp` or `quad8`? Determine this by making a table for each routine of the tolerance, the error achieved, and the number of function evaluations. (Since `quad8` expects a relative tolerance, you should divide the absolute tolerance by the exact value to get the tolerance to input to `quad8`. Also, both routines (especially `quad8`) tend to be conservative, delivering a smaller error than requested, so it is probably best to compare the number of function evaluations for a similar level of achieved error to measure efficiency.

Study the behavior of `adaptsimp` for other integrals, like $\int_0^1 x^\alpha\, dx$, for $0 < \alpha < 1$. The smaller α the more difficult (singular) this integral is.

How about $\int_0^1 |x - 1/\pi|^\alpha dx$. Here the singularity is not at an endpoint and `adaptsimp` has to "find" it.