CSE/MATH 455, Arnold Lab 11 December 10, 1997

Adaptive Quadrature

Preparation: `cp ~dna2/{adaptsimp.m,humpsd.m,showadapt.m} .`

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 relative error tolerance. Thus `quad8('humps',0,3,1.e-10)` computes the integral of the humps function over $[0, 3]$ to high accuracy.

1. 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.

2. 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 `humpsd.m` (humps with display). The script `showadapt.m` initializes the counter and graphics window and calls `adaptsimp` to integrate `humpsd`. Run it. You can vary `showadapt.m` in various ways: changing the tolerance, switching from `adaptsimp` to `quad8`.

3. For various tolerances, (say tol = 10^{-1}, 10^{-2}, ... , 10^{-8}), which is more efficient, `adaptsimp` or `quad8`? Determine this by making a table for each routine of the tolerance, the error achieved, and the number of function evaluations. Both routines tend to be conservative, delivering a smaller error than requested, so it is best to compare the number of function evaluations for a similar level of achieved error to measure efficiency.

4. Study the behavior of `adaptsimp` for the singular integral $\int_0^1 x^\alpha dx$, for $0 < \alpha < 1$. This function is not C^1 and the smaller α the more difficult (singular) this integral is.

5. Next study the performance on $\int_0^1 |x - 1/\pi|^\alpha dx$. Here the singularity is not at an endpoint and `adaptsimp` has to "find" it. How does the performance compare to the previous case?