Name: ______________________________ Section: ____________

ID#: ______________________________

CSE/MATH 455, Arnold — Lab 7 — November 22, 1996

Singular integrals

For a (simple) Matlab routine to compute Simpson's rule copy `simpson.m` from the directory `~dna2/pub/quadrature`.

Most of the theory we have developed for quadrature rules has been for integrals of smooth functions, that is, for functions which have derivatives, second deriviatives, etc., which are continuous on the entire interval of integration (including at the endpoints). This lab concerns techniques for integrating functions for which some derivatives are discontinuous somewhere on the interval.

A simple case occurs when the integrand or one of its derivatives has a simple jump discontinuity at a point in the interval. For example, consider the integral

$$\int_0^3 |\cos x| \exp(-x/10)\, dx = 1.61523492777325443\ldots. \tag{1}$$

Plot this function over the interval of integration. Note the discontinuity in the derivative at $x = \pi/2$.

Use Simpson's rule with various numbers of panels[1] to compute the integral (1) and record the error in the second column of the table:

subints	Simpson's rule error	error after splitting interval
4		
8		
16		
32		

The approach to this kind of integral is simple. Break up the integral into two pieces: $\int_0^3 = \int_0^{\pi/2} + \int_{\pi/2}^3$. In the third column of the table above put the results obtained in this way. For a fair comparison, in the first row show the result of using Simpson's rule with

[1] In my terminology, a panel is a subinterval on which you apply the simple Simpson's rule, so n panels means $2n+1$ function evaluations.

two subintervals on $[0, \pi/2]$ and two subintervals on $[\pi/2, 3]$, in the second row use four subintervals for each piece, etc.

For the integral

$$\int_0^2 \cos x \, \exp(-\sqrt{x}) \, dx = 0.49060485069310796\ldots \tag{2}$$

the difficulty is that the derivative is infinite at the left endpoint (check this!). Plot the integrand. Record the errors for Simpson's rule in this case.

subints	Simpson's rule error	err. after subtracting sing.	err. after changing variable
4			
8			
16			
32			

One approach to attacking such integrals is to "subtract off the singularity". Specifically, write

$$\int_a^b f(x)\, dx = \int_a^b [f(x) - g(x)]\, dx + \int_a^b g(x)\, dx,$$

where $g(x)$ is chosen so that (a) the function $f(x) - g(x)$ does not have an infinite derivative, and (b) the integral of $g(x)$ can be computed analytically. In our case we can take $g(x) = \exp(-\sqrt{x})$. Using the substitution $u = \sqrt{x}$ and then integrating by parts leads to $\int_0^2 g(x)\, dx = 2 - (2 + 2\sqrt{2})/\exp(\sqrt{2})$. Use Simpson's rule to evaluate the integral of $f - g$ and add in this exact value to compute the integral (2). Record the errors in column 3 of the table.

Another approach which often works is a cleverly selected change of variable. For the integral (2), the change of variable $u = \sqrt{x}$ leads to a nicer integral. Perform the change of variable, compute the resulting integral by Simpson's rule and record the errors in the second table.

Transformations such as discussed above are helpful when using adaptive quadrature routines as well. In fact the most efficient way to compute an integral is usually to do some analytical manipulations to make the integrand as nice as possible, and then use a good adaptive quadrature routine.

Use Matlab's adaptive routine `quad8` with a tolerance of zero (to get `quad8` to deliver its best possible answer) to compute the integral (2). What is the error? ___________ How many function evaluations were needed[2]? ___________

Now subtract off the singularity, use `quad8` to integrate the rest (use a tolerance of 10^{-5} and don't forget to reset `nfun` to zero before calling `quad8`), and add back the integral of the singularity. What is the error? ___________ How many function evaluations were needed? ___________

Finally compute the integral arising from the change of variable discussed above using `quad8` with a tolerance of 10^{-5}. What is the error? ___________ How many function evaluations were needed? ___________

[2] Recall the trick to count the number of function evaluations for a Matlab quadrature routine: in your function subroutine add the lines

```
global nfun
nfun = nfun + length(x);
```

At the Matlab prompt type "`global nfun`" (once is enough), and then type "`nfun = 0`" before calling the quadrature routine. After calling the quadrature routine, type "`nfun`" to show the number of function evaluations.