.,

CSS 455 Winter 2012 C. Jackels
Activity No. 13 March 6, 2012
Names (must be present):
Includes activity 13b

Part I. Composite rules for numerical quadrature.

Download the script *CompositeNCCos.m* from the Set6 examples on the web site. Execute this script with Matlab. This program uses a composite Newton Cotes (m=3) rule to evaluate the integral at the right, which we have treated previously in class using the simple NC(m) rules:

$$\int_0^{\pi/2} \cos(x)dx = \sin(x)\Big|_0^{\pi/2} = 1$$

On each iteration, the program prints out the number of integration panels (N), the approximate value of the integral and the absolute error.

1. At least in this case, does the convergence appear to be monotonic?

   Yes – the error gets smaller with each increase in number of panels.

2. Examine at least three pairs of the approximate integration errors (E_N and E_{2N}) to determine if they are approximately related by the expression with a common value of X:

   $$E_{2N} \approx \left(\frac{1}{X}\right)E_N$$

   For example, $E_{16}/E_{32} = 16.00$
   What is the average experimental value of X for the cases you chose?

   All of the X values reported were close to 16, so the average was very close.

3. Since there is a predictable relationship between successive errors (above), one may be able to use that to extrapolate to the correct answer.

   Consider the following proposed relationship, where S_N is the approximate integral with N panels and I_{exact} is the exact integral:

   $$\left(I_{exact} - S_{2N}\right) \approx \left(S_{2N} - S_N\right)\big/ Y$$

Using two choices of N, calculate an average experimental of Y:
$Y = (S_{2n} - S_N)/(I_{exact}-S_{2N})$, evaluate with N=16:
Y = (1.0000000020161288 - 1.0000000322650011)/(1.0 - 1.0000000020161288) = 14.98

All of the class values were close to 15. That value should be useful in extrapolation.

=======do not go on until after class discussion ==========================

Part II. We have seen that numerical integration of *humps* is much more challenging. Download the m-file *CompositeNCHumps.m* and execute it.

4. Using the "class value" for X above, determine at what value of *N* the integration errors for *humps* appear to be following the pattern we found above:

   The sequence of rations E_2/E_4, E_4/E_8, E_8/E_{16}, etc gives the sequence:

   1.8656 -2.4831 -2.4983 -4.2150 -74.8918 -510.8259 -26.4093 15.9624

   15.9904
   The pattern is emerging only at N = 128 panels.

5. Using the "Class value" for Y above, estimate the exact value of the *humps* integral, extrapolating from *N=128.*

$$(I_{exact}) \approx S_{2N} + (S_{2N} - S_N)/Y$$

   $I_{exact} = S_{256} + (S_{256} - S_{128})/15$ = 23.9680798021557670 + (23.9680798021557670 - 23.9680792350502760) / 15 = 23.9680798021557670 + (3.781 x 10^{-8} = **23.968079839966**
   the error is on the order of 10^{-10}

Part III.

6. Download and run *CompositeNCVariableM.m.* This run produces graphs for *cos* and *humps* functions that present integration error as a function of the number of panels and of the order of the NC integration. For each function, assume you have an approximate integral calculated with Simpson's Rule (m=3) at 30 integration panels. To improve the calculation, should you: a) double the number of panels or b) go to an m=5 NC method? Your discussion should be quantitative and provide a comparison of the two cases. How do the two cases differ?

   For cosine(x), it can be seen that the m=3 curve is pretty flat at 30 panels, so extension to 60 panels yields a modest gain, from 10^{-8} to 10^{-9}. However, dropping to the m=5 curve is a much larger change, to approximately 10^{-13} at 30 panels. So, for this easily integrated function, you are better off to go to higher order m=5 method.

   For humps(x), a much harder function to integrate, extension of (m=3) from 30 to 60 panels will drop the error from 10^{-2} to approximately 10^{-4}. However, dropping to m=5 at 30 panels only drops the error to 10^{-3}. In this case, you are better off to double the number of panels.

Part IV

Using the approximation:

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h}$$

estimate the value of cos(π/2) and report the error for your assigned value of *h*. The exact answer is -0.1
For the h values:

h =

Columns 1 through 4

0.100000000000000   0.010000000000000   0.001000000000000   0.000100000000000

Columns 5 through 8

0.000010000000000   0.000001000000000   0.000000100000000   0.000000010000000

Columns 9 through 10

0.000000001000000   0.000000000100000

The approximate derivative at these *h* values are:
fprime =

Columns 1 through 4

-0.998334166468282   -0.999983333416667   -0.999999833333232   -0.999999998333223

Columns 5 through 8

-0.999999999989885   -0.999999999917567   -1.000000000583866   -0.999999993922529

Columns 9 through 10

-1.000000082740371   -1.000000082740371

error =

Columns 1 through 4

0.001665833531718   0.000016666583333   0.000000166666768   0.000000001666777

Columns 5 through 8

0.000000000010115   0.000000000082433   -0.000000000583866   0.000000006077471

Columns 9 through 10

-0.000000082740371   -0.000000082740371

As expected, the error gets smaller as h decreases until $h = 10^{-5}$ and then starts increasing due to numerical problems.