Problem set 9

[Revised Jan 29, 9:21 PM]

Numerical Integration

For all questions on this problem set we consider evaluating the definite integral

$$I = \int_0^3 \left[\sin(0.5x^3) + 1.0\right]\, dx$$

for which the exact value of I is

```
I = 3.5158553705912787e+00
```

The integrand can be evaluated using the following Python function

```
import numpy

def integrand_function(x):
    return numpy.sin(0.5*x**3) + 1.0
```

---

1. Complete the Python function `integrate_midpoint_rule()` found in `midpoint.py` to perform the midpoint integration rule You will call the function `integrate_midpoint_rule()` like this

   ```
   I_m = integrate_midpoint_rule(  \
               0.0, 3.0, \
               ncells, integrand_function)
   ```

   where `ncells` is the number of segments defined over the domain $[0, 3]$, and `integrand_function` is a Python function defining the integrand.

   - Using your function `midpoint()`, use the midpoint rule to estimate I when using `ncells = 80`, `ncells = 160`, `ncells = 320`. Call the results `I_80`,

`I_160`, `I_320`.
  - Compute the error assocaited with midpoint method (using the exact value of I provided at the top of this problem set) with `ncells = 80`, `ncells = 160`, `ncells = 320`. Verify the order of accuracy of the midpoint rule is $O(h^2)$. Explain your reasoning.
  - Use `I_80` and `I_160` with Richardson extrapolation to obtain a new approximation for I. Call this `I_160_rich`. Compute the error of `I_160_rich`.
  - Use `I_160` and `I_320` with Richardson extrapolation to obtain aanother approximation for I. Call this `I_320_rich`. Compute the error of `I_320_rich`.
  - Verify that the Richardson extrapolation procedure is $O(h^4)$ accurate. Explain your reasoning.

2. Use Gaussian quadrature (`scipy.integrate.fixed_quad()`) with order (`n`) = 5, 10, 15, 20. Compute the error for each estimate of I obtained.

3. Use the trapezoid rule with 320 points (`scipy.integrate.trapezoid()`). Report the value of I_m obtained and compute the error of the approximation.

4. Use Simpson's rule with 320 points (`scipy.integrate.simpson()`). Report the value of I_m obtained and compute the error of the approximation.

5. Use the adaptive quadrature method (`scipy.integrate.quad()`). Report the value of I_m obtained and compute the error of the approximation.