Ph20.2 - Introduction to Numerical Techniques: Numerical Integration

Cassidy Yang

January 29, 2016

1 The Assignment

The corresponding Python code is attached in a separate document.

1. Derive extended Simpson's formula.

   By elementary definite integration and also an extension of Eq. (4) in the assignment notes, we get

   $$I = f(a)H + f'(a)\frac{H^2}{2!} + f''(a)\frac{H^3}{3!} + f^{(3)}(a)\frac{H^4}{4!} + f^{(4)}(\eta)\frac{H^5}{5!} + ... \text{ with } \eta \in [a,b]. \tag{1}$$

   We can can use *Simpson's rule* from Eq. (8) in the assignment notes and reproduced below where $c = \dfrac{(a+b)}{2}$

   $$I \simeq = I_{simp} \equiv H\left(\frac{f(a)}{6} + 4\frac{f(c)}{6} + \frac{4f(b)}{6}\right). \tag{2}$$

   Additionally, we can write out $f(x)$ as a Taylor sum centered at a:

   $$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \frac{f^{(4)}(\eta)}{4!}(x-\eta)^4 \text{ with } x, \eta \in [a,b]. \tag{3}$$

   Using Eq. (3) to approximate $f(b)$ and $f(c)$ so we can rewrite Eq. (2) and find the error term.

   $$\begin{aligned} I_{simp} &= f(a)\frac{H}{6} + \left[f(a) + f'(a)\frac{H}{2} + \frac{f''(a)}{2}\left(\frac{H}{2}\right)^2 + \frac{f^{(3)}(a)}{6}\left(\frac{H}{2}\right)^3 + \frac{f^{(4)}(a)}{24}\left(\frac{H}{2}\right)^4\right]\frac{2H}{3} \\ &\quad + \left[f(a) + f'(a)H + \frac{f''(a)}{2}H^2 + \frac{f^{(3)}(a)}{6}H^3 + \frac{f^{(4)}(a)}{24}H^4\right]\frac{H}{6} \\ &= f(a)H + f'(a)\left(\frac{H^2}{3}\right) + \frac{H^2}{6} + f''(a)\left(\frac{H^3}{12} + \frac{H^3}{12}\right) + f^{(3)}(a)\left(\frac{H^4}{72} + \frac{H^4}{36}\right) + f^{(4)}(\eta)\left(\frac{H^5}{576} + \frac{H^5}{144}\right) \\ &= f(a)H + f'(a)\frac{H^2}{2!} + f''(a)\frac{H^3}{3!} + f^{(3)}(a)\frac{H^4}{4!} + f^{(4)}(\eta)\frac{5H^5}{576} \end{aligned} \tag{4}$$

   $$I_{simp} - I = f^{(4)}(\eta)\frac{5H^5}{576} - f^{(4)}(\eta)\frac{H^5}{120} = f^{(4)}(\eta)\frac{H^5}{2880} \tag{5}$$

   Hence, Eq. 5 tells us that the *Simpson's rule* is locally of fifth order in H (ie the local error is $O(H^5)$ since we can rewrite it as

   $$I = I_1 + O(H^5). \tag{6}$$

We can improve the estimation of the integral by dividing the interval $[a, b]$ into N subintervals. We define $h_N = (b - a)/N$, and $x_0 = a, x_1 = a + h_N, ...x_N = b$. We can write the *extended Simpson's formula.*

$$\begin{aligned}\int_a^b f(x)dx &= \int_{x_0}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + ... + \int_{x_{N-1}}^{x_N} f(x)dx \\ &= h_N\left[\frac{f(x_0)}{6} + \frac{2f(\frac{(x_0+x_1)}{2})}{3} + \frac{f(x_1)}{6}\right] + ... + h_N\left[\frac{f(x_{N-1})}{6} + \frac{2f(\frac{x_{N-1}+x_N}{2})}{3} + \frac{f(x_N)}{6}\right] \\ &= h_N\left[\frac{f(x_0)}{6} + \frac{2f(\frac{(x_0+x_1)}{2})}{3} + \frac{f(x_1)}{3} + \frac{2f(\frac{x_1+x_2}{2})}{3} + \frac{f(x_2)}{3} + ... + \frac{f(x_N)}{6}\right] \qquad (7)\end{aligned}$$

The global error in this case is $O(h_N^4)$, since

$$-f^{(4)}(\xi)\frac{h_N^5}{2880} \cdot N = -(b-a)f^{(4)}(\xi)\frac{h_N^4}{2880} \text{ with } \xi \in [a, b]. \qquad (8)$$

2. See attached code for trapezoidal extended rule implementation.

3. See attached code for Simpson's extended rule implementation.

4. Evaluation of $\int_0^1 e^x dx$.

   The value of the integral is $e - 1$, which is approximated using the python "`np.e - 1`". Figure 1 shows convergence of the trapezoidal and Simpson's rule in a log-log plot. The Simpson's rule seems to converge at a much faster rate (seen as a steeper slope). Note that at large values of N, there seems to be some error.

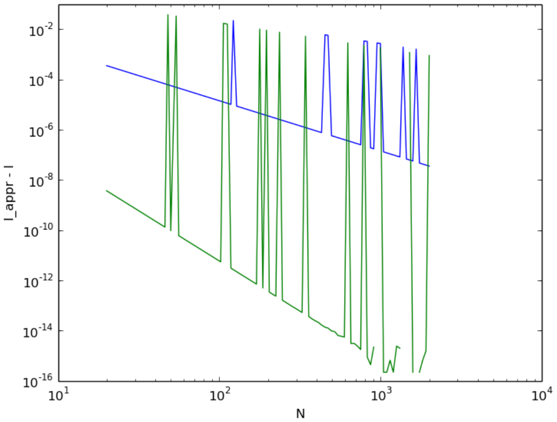

Figure 1: Convergence rate for the trapezoidal formula (blue) and Simpson's rule (green)

   The errors result due to some rounding error present while making numerical calculations with float data type. This can be remedied by using the Decimal object instead from the decimal package. Figure 2 shows the corrected convergence rate.

5. See attached code for refined Simpson's formula.

   The function suitably refines the extended Simpson's formula for suitable number of intervals, N, until it has a desired accuracy. It is tested on the functions e^x and x^2 for a difference between successive approximations of less than 10^{-6}. Table 1 shows the convergence of e^x for increasing values of N. The approximation for the function x^2 is immediately within the desired accuracy for $N = 1$ and no further refinement is necessary.

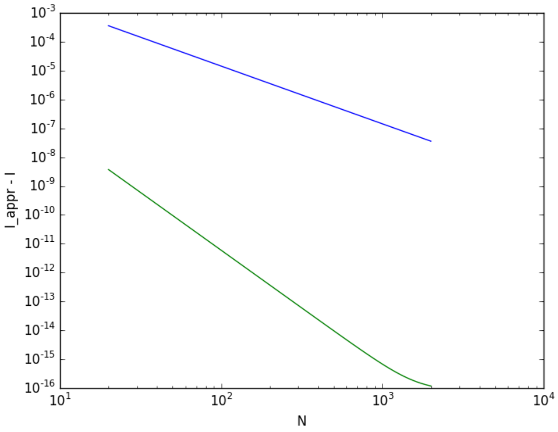

Figure 2: Corrected Convergence rate for trapezoidal (blue) and Simpson's (green)

Table 1: Convergence to e^x

N	I_{simp}	$\left\lvert \frac{I_{simp(N)} - I_{simp(2N)}}{I_{simp(N)}} \right\rvert$
1	1.71886115188	0.000315505388119
2	1.71831884192	2.01867203012e-05
4	1.7182841547	1.26908462684e-06
8	1.71828197405	7.94340638826e-08

6. Using `scipy.integrate.quad` and `scipy.integrate.romberg`

   The `scipy.integrate.quad` and `scipy.integrate.romberg` functions are able to calculate the integral much faster with a much smaller error than all of our previous functions.