General formula to calculate R_{jk} in Romberg's method:

$$R_{jk} = \frac{4^{k-1} R_{j,k-1} - R_{j-1,k-1}}{4^{k-1} - 1} = R_{j,k-1} + \frac{R_{j,k-1} - R_{j-1,k-1}}{4^{k-1} - 1}$$

For algorithm and Matlab Code see book (Sauer, T.)

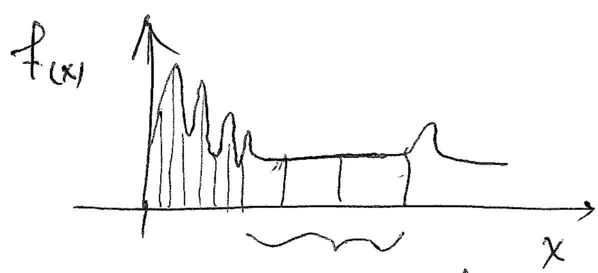
Notes:

So far, we have learned how to estimate an integration using numerical methods that use equal step sizes.

There two limitations with applying equal step sizes:

1. Comparing approximation error with an error tolerance is difficult. Because error formula involves higher derivatives which may be complicated, which makes it difficult to calculate the error and compare it with tolerance.

2. Rapidly varying functions will require smaller step sizes. when the function varies at small ~~steps~~ parts of the domain.



Thus, we will find out what step size is appropriate for a specific subinterval.

Adaptive Quadrature (will solve both problems). ②

Idea is to refine subintervals adaptively, over the integration domain.

For every sub-interval, the method checks:

if the estimation error is larger than the required tolerance then that interval is sub-divided and quadrature is applied on both $\overset{\text{half}}{\wedge}$ sub-intervals.

we will use Trapezoid and Simpson's rules:

$\boxed{1}$ Adaptive quadrature using Trapezoid rule:

$$\text{Recall: } \int_a^b f(x)\,dx \simeq \underbrace{\frac{h}{2}\left(y_a + y_b\right)} - h^3 \frac{f''(c_0)}{12} = \underbrace{S_{[a,b]}} - h^3 \frac{f''(c_0)}{12} \quad (1)$$

c_0 in $[a,b]$

Let c be the midpoint of $[a,b]$, we can write Trapezoid rule for both half-intervals: $[a,c]$ and $[c,b]$:

$$\int_a^b f(x)\,dx \simeq S_{[a,c]} - \frac{h^3}{8}\frac{f''(c_1)}{12} + S_{[c,b]} - \frac{h^3}{8}\frac{f''(c_2)}{12}$$

c_1 in $[a,c]$ $\qquad$ c_2 in $[c,b]$

$$\simeq S_{[a,c]} + S_{[c,b]} - \frac{h^3}{4}\frac{f''(c_3)}{12} \quad (2)$$

(1), (2)
Error difference: $S_{[a,b]} - \left(S_{[a,c]} + S_{[c,b]}\right) = h^3\frac{f''(c_0)}{12} - \frac{h^3}{4}\frac{f''(c_3)}{12}$

$$\simeq \frac{3}{4}h^3\frac{f''(c_3)}{12}$$

$$f''(c_0) \simeq f''(c_3)$$

Error differen: by dividing the interval, the estimation error drops 3 times. ③

For each halfs estimation error drops by a factor of 8.

Using $S_{[a,b]} - \left(S_{[a,c]} + S_{[c,b]}\right)$ we can check if the approximation tolerance is met.

For a sufficient number of halvings, the tolerance will be met.

See book: algorithm and Matlab Cod.

$\boxed{2}$ Adaptive quadrature using Simpson's rule:

Recall: $$\int_a^b f(x)\,dx \simeq \frac{h}{3}\left(y_a + 4y_c + y_b\right) - \frac{h^5}{90} f^{(4)}(c_0) = S_{[a,b]} - \frac{h^5}{90} f^{(4)}(c_0) \quad (1)$$

c_0 in $[a,b]$

Let c be the midpoint of $[a,b]$, we can use Simpson's rule, for both half-intervals $[a,c]$ and $[c,b]$

$$\int_a^b f(x)\,dx \simeq S_{[a,c]} - \frac{h^5}{32}\frac{f^{(4)}(c_1)}{90} + S_{[c,b]} - \frac{h^5}{32}\frac{f^{(4)}(c_2)}{90}$$

c_1 in $[a,c]$ $\qquad$ c_2 in $[c,b]$

$$\simeq S_{[a,c]} + S_{[c,b]} - \frac{h^5}{16}\frac{f^{(4)}(c_3)}{90} \quad (2)$$

From (1), (2): $$S_{[a,b]} - \left(S_{[a,c]} + S_{[c,b]}\right) = \frac{h^5 f^{(4)}(c_0)}{90} - \frac{h^5}{16}\cdot\frac{f^{(4)}(c_3)}{90}$$

$\Big\downarrow$ assuming $f^{(4)}(c_0) \simeq f^{(4)}(c_3)$

$$\simeq \frac{15}{16} h^5 \frac{f^{4}(c_3)}{90}$$

For Code see the book.

This means by dividing interval into two ~~halves~~ panels, the estimation error drops 15 times. Thus, adaptive quadrature using Simpson's rule would require less subdivisions and converges to integral solution faster, than when it uses Trapezoid rule.

Gaussian Quadrature.

Second method that relaxes the evenly spaced points requirement.