Home Assignments in Numerical Methods

Vasily Arzhanov

HA06 (20p)

Preamble. Integration is of paramount importance in science in general and in engineering applications in particular, for example

$$\Delta\mathbf{p} = \int_{t}^{t+\Delta t} \mathbf{F}(\tau)\mathrm{d}\tau; \qquad \phi_g(\mathbf{r}) \equiv \frac{1}{E_g - E_{g+1}} \int_{E_{g+1}}^{E_g} \phi(\mathbf{r}, E)\mathrm{d}E$$

Georg Friedrich Bernhard Riemann, 1826 – 1866, was the first to give a strict definition of the definite integral. It begins with a partition of an interval [a, b], which is determined by a finite sequence of real numbers x_i and ξ_i of the form:

$$P(x_i, \xi_i) \equiv \left\{ x_i, \xi_i \middle| a = x_0 < x_1 < \ldots < x_n = b; \quad \xi_i \in [x_{i-1}, x_i] \right\}$$

The definite integral is then defined by

$$\int_a^b f(x)\mathrm{d}x \equiv \lim_{h\to 0} \sum_{i=1}^{n} f(\xi_i)(x_i - x_{i-1}) \qquad h \equiv \max_{1\le i\le n}(x_i - x_{i-1})$$

Any finite sum of the above form is referred to as a Riemann sum. Selecting the numbers ξ_i that minimise $f(x)$ on every subinterval [x_{i-1},x_i], we arrive at the so called the lower Riemann sum. The upper Riemann sum is defined similarly. Their importance stems from the fact that

$$\sum_{i=1}^{n} \min_{[x_{i-1}, x_i]} f(\xi)(x_i - x_{i-1}) \le \int_a^b f(x)\mathrm{d}x \le \sum_{i=1}^{n} \max_{[x_{i-1}, x_i]} f(\xi)(x_i - x_{i-1})$$

Numerical integration is used to evaluate a definite integral when there is no closed-form expression for the definite integral. A formula or algorithm that gives a numerical, generally approximate, value of the definite integral in question is called quadrature.

$$I(f) \equiv \int_a^b f(x)\mathrm{d}x \approx Q(f); \qquad E(f) \equiv I(f) - Q(f)$$

Some authors restrict this terminology for univariate integrands, $f(x)$, while reserving "cubature" for multi-dimensional integration. Sometimes, it is important to stress the integration limits, $E(f) \equiv E(f, a, b)$. One example of a quadrature that is usually credited to Thomas Simpson, 1710–1761, divides the interval [a, b] into $N = 3n$ uniform subintervals thus covering it with $N + 1$ evenly spaced points (nodes), x_i:

$$h = (b-a)/N; \quad x_i = a + i\cdot h; \quad i = 0, 1, \cdots, N$$

This quadrature is referred to as the composite Simpson's 3/8 rule. It is defined as

$$I(f) \approx Q_{3/8}(f) \equiv \frac{3}{8}h\left[f(x_0) + 3\sum_{i=1, i\ne 3k}^{N-1} f(x_i) + 2\sum_{j=1}^{N/3-1} f(x_{3j}) + f(x_n) \right]$$

The error associated with this quadrature is

$$E_{3/8}(f, a, b) = -\frac{h^4}{80}(b-a) f^{(4)}(\xi); \qquad \xi \in [a, b]$$

Typically, we know very little about ξ and hence cannot use this representation directly. However, it suggests the following upper estimate

$$\left| E_{3/8}(f, a, b) \right| \le \frac{h^4}{80}(b-a) \max_{\xi\in[a,b]} \left| f^{(4)}(\xi) \right|$$

Exercise 1 (3p). As a rule, the error term of a quadrature formula involves a derivative of the integrand. If the integrand is given by a "well-defined" formula, one can use built-in symbolic libraries. To this end, read corresponding documentation about symbolic objects. Then do the following

a) Define and print a symbolic object that represents the function $f(t) = 1/\ln t$. Make your conclusion about monotonicity of this function when $2 \le t < \infty$

b) Symbolically calculate and print $f''(t)$; then numerically evaluate the second derivative at $t = 2$. Is $f''(t)$ monotonic on the same interval, $2 \le t < \infty$?

c) Symbolically calculate and print $f^{(4)}(t)$; then numerically evaluate the fourth derivative at $t = 2$. Give your conclusion whether or not $f^{(4)}(t)$ is monotonic on the same interval, $2 \le t < \infty$.

Selecting the numbers ξ_i at the left/right end of the subinterval $[x_{i-1}, x_i]$, we arrive at the so called left/right Riemann sums defined as follows

$$L(f, x_i) \equiv \sum_{i=1}^{n} f(x_{i-1})(x_i - x_{i-1}) \qquad R(f, x_i) \equiv \sum_{i=1}^{n} f(x_i)(x_i - x_{i-1})$$

For a monotonically increasing on $[a, b]$ function, $f(x)$, the left/right Riemann sum coincides with the lower/upper Riemann sum. For a monotonically decreasing function, the situation is just the opposite. In either case, the definite integral is squeezed between these sums i.e., when a function, $f(x)$, is monotonic on $[a, b]$, it holds

$$\min\{L(f, x_i), R(f, x_i)\} \le \int_a^b f(x)\mathrm{d}x \le \max\{L(f, x_i), R(f, x_i)\}.$$

A nice property of this pair of quadrature rules is that the absolute error estimate, $|R - L|$, does not involve a derivative of the integrand $f(x)$.

Exercise 2 (1p). Armed with this piece of knowledge, carry out the following steps.

a) Show that for a monotonically decreasing function $f(x)$ over an interval $[a, b]$, it holds, $R(f, x_i) \le \int_a^b f(x)dx \le L(f, x_i)$, which gives an uncertainty interval.

b) Prove that for a monotonically decreasing function with n uniform subintervals, the difference between the left (in this case – upper) Riemann sum and the right (in this case – lower) Riemann sum is given by $L - R = [f(a) - f(b)] \cdot (b - a)/n$.

The logarithmic integral function or simply integral logarithm, li(x), and the offset logarithmic function also known as Eulerian logarithmic integral, Li(x), are special functions defined as

$$\mathrm{li}(x) \equiv \int_0^x \frac{\mathrm{d}t}{\ln t} \text{ and } \mathrm{Li}(x) \equiv \int_2^x \frac{\mathrm{d}t}{\ln t} = \mathrm{li}(x) - \mathrm{li}(2)$$

The integral logarithm, li(x), requires a special treatment of the singularity at $t = 1$ whereas the offset function, Li(x), has the advantage of avoiding the singularity in the domain of integration.

Exercise 3 (3p) continues the previous exercise with the following steps.

a) Both MATLAB and Python provide these logarithmic functions. Evaluate Li(200) by an appropriate standard function. Write a computer code that calculates the left, L, and right, R, Riemann quadrature using n evenly spaced subintervals. Based on the result in Exercise 2 b), evaluate the number of subintervals n needed to approximate Li(200) by L or R, with 3 decimal places. Report n, R, Li(200) and L using at least 6 decimal digits.

b) Treating Li(200) as exact, report the actual absolute and relative errors contained in the left, L, and right, R, Riemann quadrature. It is also logical to average these two approximations, $T = (L+R)/2$. Report the absolute and relative errors in T.

c) Interestingly, the Eulerian logarithmic function, Li(x), approximates the number of primes in the interval $[1, x]$. Moreover, it is proven that $\lim_{x\to\infty} \pi(x)/\mathrm{Li}(x) = 1$, where $\pi(x)$ is the prime-counting function that gives how many prime numbers there are in $[1, x]$, for example, $\pi(1) = 0$, $\pi(2) = 1$, $\pi(5.5) = 3$. Find this function in a standard library, evaluate $\pi(200)$ and compare with quadrature T that approximates Li(200).

Selecting the numbers ξ_i at the midpoint of the subinterval $[x_{i-1},x_i]$, we arrive at the so called midpoint quadrature. In case of uniform subintervals, the composite midpoint quadrature assumes an extremely simple form

$$h = (b-a)/n; \quad \xi_i = a + \left(i + 1/2\right)h; \quad M\left(f,\xi_i\right) = h\sum_{i=0}^{n-1} f\left(\xi_i\right).$$

The error estimate is known to be

$$\left|E_M(f)\right| \le \frac{\max_{a\le x\le b} f''(x)\cdot(b-a)}{24}h^2 = \frac{\left\|f''\right\|_\infty (b-a)^3}{24n^2}$$

Exercise 4 (1p). Write a computer function that implements the midpoint quadrature M. Estimate the number of subintervals, n, needed to approximate Li(200) with 3 decimal places using the midpoint quadrature i.e., guarantee that $\left|E_M(f)\right| \le 10^{-3}$. Exercise 1 might be helpful here in evaluating $\left\|f''\right\|_\infty$. Then do the following.

a) Report the norm of the second derivative, $\left\|f''\right\|_\infty$, on the interval [2,200] as well as the number of subintervals, n.

b) Report the midpoint approximation of Li(200) together with the absolute and relative errors.

Averaging the left and right Riemann sums produces another popular quadrature known as the trapezoidal, trapezoid or trapezium rule. Indeed, for a small interval $[\alpha, \beta]$, $L = f(\alpha)(\beta-\alpha)$, $R = f(\beta)(\beta-\alpha)$, which yields

$$T \equiv \frac{L+R}{2} = \frac{f(\alpha)+f(\beta)}{2}(\beta-\alpha)$$

It is interesting to note, the error estimate for the Midpoint quadrature is approximately twice as better than that for the trapezoid rule.

$$E_M = \frac{(\beta-\alpha)^3}{24}f''(\zeta) + O\left((\beta-\alpha)^4\right) \quad \text{and} \quad E_T = -\frac{(\beta-\alpha)^3}{12}f''(\zeta) + O\left((\beta-\alpha)^4\right)$$

It is not difficult to show that the linear combination $S \equiv (2M+T)/3$ eliminates the leading terms in the corresponding error estimates thus giving a more accurate quadrature. It is known as Simpson's rule. In case of uniform subintervals,

$$h = \frac{b-a}{n}; \quad x_i = a + \frac{h}{2} i; \quad i = 0,1,2,\cdots,2n; \quad x_0 = a; \quad x_{2n} = b.$$

composite Simpson's rule reads as

$$S(f, x_i) = \frac{h}{6}\left[f(x_0) + 2\sum_{i=1}^{n-1} f(x_{2i}) + 4\sum_{i=0}^{n-1} f(x_{2i+1}) + f(x_{2n}) \right]$$

The error estimate is proven to be

$$|E_S(f)| \le \frac{\max\limits_{a \le x \le b} f^{(4)}(x) \cdot (b-a)}{2880} h^4 = \frac{\left\| f^{(4)} \right\|_\infty (b-a)^5}{2880 n^4}.$$

Write a computer function that implements Simpson's quadrature. It will be used to evaluate improper integrals of a Gaussian function

$$\int_0^\infty e^{-ax^2} \mathrm{d}x = 0.5\sqrt{\pi/a}$$

For simplicity, we take $a = 1$ thus expecting the answer $\sqrt{\pi}/2$. Numerically evaluating improper integrals is not an easy task; it calls for some analytic work. Often a suitable change of variables may give a satisfactory result. Here we will employ an obvious and straightforward approach of splitting the integration interval into 2 regions

$$\int_0^\infty e^{-x^2} dx = \int_0^A e^{-x^2} dx + \int_A^\infty e^{-x^2} dx$$

Let the absolute tolerance be given by a small number ε. By noting that $x^2 > x$ when $x > 1$, we select first $A > 1$ such that

$$0 < \int_A^\infty e^{-x^2} dx < \int_A^\infty e^{-x} dx \le \frac{\varepsilon}{2}$$

Secondly, we select the number of subintervals n in Simpson's rule that guarantees

$$\left| \int_0^A e^{-x^2} \mathrm{d}x - Q(f, 0, A) \right| = |E_S(f, 0, A)| \le \frac{\left\| f^{(4)} \right\|_\infty (A-0)^5}{2880 n^4} \le \frac{\varepsilon}{2}$$

Here, $f(x) = e^{-x^2}$. In doing so, we ensure that the total absolute error does not exceed the predefined tolerance, ε.

Exercise 5 (3p). Armed with the previous piece of knowledge, do the following.

a) By analysing the error estimate, identify the class of functions, for which Simpson's algorithm gives the exact answer. Test your computer code using such a function.

b) Set $\varepsilon = 10^{-3}$ and find A such that $\int_A^\infty e^{-x} dx \le \frac{\varepsilon}{2}$

c) Define a symbolic object that represents the Gaussian function, $G(x) = e^{-x^2}$; calculate symbolically the fourth derivative, $G4(x) = G^{(4)}(x)$; simplify it using a built-in function and finally print it.

Exercise 6 (3p) continues the previous exercise. It is recalled $f(x) = \exp(-x^2)$.

a) Plot the fourth derivative on the interval [0,5] and visually identify the point at which $\left|G4(t)\right|$ assumes the maximum then evaluate $G4(t)$ at this point thus finding $\left\|f^{(4)}\right\|_\infty$ in the error estimate.
b) Evaluate the number of subintervals on $[0, A]$ in the Simpson's rule that guarantees $\left|E_S(f,0,A)\right| \le \varepsilon/2$.
c) Evaluate numerically the integral in question and report the actual absolute and relative errors.

The Newton-Cotes formulas, also called the Newton-Cotes quadrature rules or simply Newton-Cotes rules, belong to a family of formulas for numerical integration based on evaluating the integrand at equally spaced nodes. They are named after Isaac Newton, 1642–1726, and Roger Cotes, 1682–1716. The Newton-Cotes formulas are useful when the integrand is given as tabulated values at equidistant points.

A very general approach to construct a numerical quadrature comes through two steps: (a) divide the (global) integration interval, say $[a, b]$, into subintervals not necessarily of equal widths; and (b) apply a quadrature rule on every (local) subinterval. Step (a) may be referred to as the composite principle, the essence of which is based on the additivity property of the definite integral. If (global) nodes X_j (note uppercase) cover the global interval $[a, b]$ such that $X_0 = a$ and $X_n = b$, then

$$\int_a^b f(x)\mathrm{d}x = \sum_{j=0}^{n-1} \int_{X_j}^{X_{j+1}} f(x)\mathrm{d}x$$

In case of equally spaced global nodes, we have

$$X_j = a + j\cdot H;\ \ j = 0,1,\cdots,n; \quad H = (b-a)/n\,.$$

At step (b), we select a particular Newton-Cotes quadrature, the essence of which is to partition every local subinterval $X_j \equiv \alpha \le x \le \beta \equiv X_{j+1}$ (for some j) into m panels with evenly spaced (local) nodes (note lowercase)

$$x_i = \alpha + i\cdot h;\ \ i = 0,1,\cdots,m; \quad h = (\beta-\alpha)/m\,.$$

Then the Lagrange polynomial, $L(x)$, that approximates the integrand $f(x)$ at $[\alpha, \beta]$, is defined through cardinal functions $\ell_i(x)$ (also polynomials) as

$$f(x) \approx L(x) = \sum_{i=0}^{m} f(x_i)\ell_i(x)$$

It is worth noting, the polynomial $L(x)$ coincides with $f(x)$ at $m + 1$ nodes, $L(x_i) = f(x_i)$, thus having a degree of m. The Newton-Cotes quadrature is then given by

$$\int_\alpha^\beta f(x)\mathrm{d}x \approx \int_\alpha^\beta L(x)\mathrm{d}x = \int_\alpha^\beta \sum_{i=0}^{m} f(x_i)\ell_i(x)\mathrm{d}x = \sum_{i=0}^{m} w_i f(x_i) \qquad w_i = \int_\alpha^\beta \ell_i(x)\mathrm{d}x$$

It turns out that the weights, w_i, do not depend on the integration interval $[\alpha, \beta]$ when the local nodes are equidistant, and thus may be precalculated as follows

$$w_i = h\int_0^m \varphi_i(t)dt; \qquad \varphi_i(t) \equiv \prod_{\substack{k=0\\k\ne i}}^{m} \frac{t-k}{i-k}.$$

The number of subintervals, n, can be arbitrarily big whereas the number of panels, m, must be reasonably low otherwise it results in a polynomial of high degree, $L(x)$, which is known to suffer from catastrophic Runge's phenomenon.

Newton-Cotes quadrature formulas come in two flavours. A quadrature is said to be closed if it involves the values of $f(x)$ at the ends of the local interval otherwise it is said to be open. Typically, the closed form supersedes the open variant. Nevertheless, open quadrature rules do have applications in integrating functions with singularities as well as in the numerical solution of ordinary differential equations.

The purpose of the next exercise is to write a computer code that implements a composite rule based on the three-point Newton-Cotes open formula, also known as Milne's rule, when every subinterval is split into $m = 4$ panels and only 3 internal nodes are involved

$$\int_{\alpha}^{\beta} f(x)\mathrm{d}x = \frac{4}{3}h\left[2f_1 - f_2 + 2f_3\right] + \frac{28}{90}h^5 f^{(4)}(\xi)$$

Here, in case of evenly spaced global nodes, $\beta - \alpha = X_{j+1} - X_j = H = (b-a)/n$,

$$f_i \equiv f(x_i);\quad x_i = \alpha + i\cdot h;\ i = 1,2,3;\quad h = \frac{\beta - \alpha}{4} = \frac{b-a}{4n}.$$

As seen, the local absolute error is $O(h^5)$. Clearly, the global absolute error is one order less, $O(h^4)$, hence when doubling the number of subintervals n, we expect improving in the global absolute error by a factor of 16.

Exercise 7 (4p). Armed with the previous piece of knowledge, carry out the following steps.

a) By analysing the error formula, identify the class of functions, for which Milne's rule gives the exact answer. Test your computer code using a suitable function.

b) The function $\mathrm{Si}(x) \equiv \int_0^x \sin t/t\,\mathrm{d}t$ is an important special function known as the sine integral. The Milne's quadrature might be convenient here because it avoids a problematic point at $t = 0$. Evaluate Si(1) using the Milne's composite rule with only $n = 2$ subintervals thus the total number of nodes will be $2\times3 = 6$. Compare this value with that given by a built-in function and report the absolute and relative errors.

c) The Fresnel integrals, S(x) and C(x), are two transcendental functions named after Augustin-Jean Fresnel (1788 – 1827) that are used in optics and closely related to the error function, erf(x). They are defined through, $\mathrm{S}(x) \equiv \int_0^x \sin(t^2)\mathrm{d}t$ and $\mathrm{C}(x) \equiv \int_0^x \cos(t^2)\mathrm{d}t$. Evaluate $\mathrm{S}\left(\sqrt{\pi}\right)$ using the Milne's composite rule with $n = 8$ subintervals. Compare this value with that given by a built-in function and report the absolute and relative errors. Denote this absolute error as errA1. Re-evaluate $\mathrm{S}\left(\sqrt{\pi}\right)$ using the Milne's composite rule with $n = 16$ subintervals. Report the absolute and relative errors, denote this absolute error as errA2 and report the ratio errA1/errA2. Does it comply with our expectations?

d) It is easy to establish $\int_0^1 \sqrt{x}dx = 2/3$. Evaluate this definite integral using Milne's composite rule first with $n = 16$ subintervals and then with twice more, $n = 32$, subintervals. Report the ratio errA1/errA2 and explain why it does not follow our expectations as we observed in c).

Exercise 8. The Gaussian quadrature chooses the nodes and weights to yield an exact result for polynomials of as high degree as possible. Suppose we want to determine two nodes, let's say x_1 and x_2, and two weights, w_1 and w_2, such that the quadrature

$$I(f) \equiv \int_{-1}^{1} f(x)dx \approx w_1 f(x_1) + w_2 f(x_2) \equiv Q(f)$$

returns the exact result for any polynomial, $p_n(x)$, of as high degree as possible. Find the nodes and weights as well as the degree of exactness n.