LAB1: ADAPTIVE NUMERICAL INTEGRATION
DUE: FRIDAY, FEBRUARY 28, 23:59

1. Introduction

In a numerical integration scheme the integral is replaced by a finite sum:

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^n f(x_i)w_i := Q. \tag{1}$$

Here $w_0, w_2, \ldots w_n$ are real numbers, called *quadrature weights*. The sets of *quadrature points* points $\{x_i\}_{i=1}^n$ in $[a,b]$ and weights $\{w_i\}_{i=1}^n$ together form a *quadrature rule*. While an exact evaluation of the left hand side (1) would require knowing the antiderivative of f, the right hand side only requires that the values of f are known at the quadrature points.

First note that if we have a quadrature rule for one interval $[a,b]$, then we can easily adapt it to any other interval $[\tilde{a},\tilde{b}]$ by transformation of the variables. The mapping $\phi : x \mapsto \frac{b-x}{b-a}\tilde{a} + \frac{x-a}{b-a}\tilde{b}$ maps the interval $[a,b]$ onto $[\tilde{a},\tilde{b}]$, hence

$$\begin{aligned}
\int_{\tilde{a}}^{\tilde{b}} f(\tilde{x})\,d\tilde{x} &= \int_a^b f\circ\phi(x)\phi'(x)\,dx \\
&= \int_a^b f\left(\frac{b-x}{b-a}\tilde{a} + \frac{x-a}{b-a}\tilde{b}\right)\left(\frac{\tilde{b}-\tilde{a}}{b-a}\right)dx \\
&\approx \sum_{i=1}^n f\left(\frac{b-x_i}{b-a}\tilde{a} + \frac{x_i-a}{b-a}\tilde{b}\right)\left(\frac{\tilde{b}-\tilde{a}}{b-a}\right)w_i.
\end{aligned} \tag{2}$$

This shows that a quadrature rule for the interval $[\tilde{a},\tilde{b}]$ with points $\{\tilde{x}_i\}_{i=1}^n$ and weights $\{\tilde{w}_i\}_{i=1}^n$ is obtained by setting

$$\tilde{x}_i = \frac{b-x_i}{b-a}\tilde{a} + \frac{x_i-a}{b-a}\tilde{b}, \tag{3}$$

$$\tilde{w}_i = \frac{\tilde{b}-\tilde{a}}{b-a}w_i. \tag{4}$$

That is, the quadrature points in $[\tilde{a},\tilde{b}]$ are distributed with the same relative distances as those in $[a,b]$, and the weights are simply scaled by the ratio of the interval lengths.

We say that a quadrature of the form (1) is of order $p \in \mathbb{N}_0$, if for all $f \in C^p[a,b]$ the estimate

$$|e| := \left|\int_a^b f(x)\,\mathrm{d}x - \sum_{j=1}^n w_j f(x_j)\right| = \left|\int_a^b f(x)\,\mathrm{d}x - Q\right| \le c \max_{x\in[a,b]}\left|f^{(p)}(x)\right| \tag{5}$$

holds for some $c > 0$ independent of f. We saw in class that if we denote the set of real polynomials of degree at most m by $\mathbb{P}_m$, then a quadrature of the form (1) is of order $p \geq 1$ if and only if it is exact for all $f \in \mathbb{P}_{p-1}$. Simpon's rule; that is, when

$$\text{points:} \qquad x_1 = a, \quad x_2 = \frac{b+a}{2}, \quad x_3 = b$$
$$\text{weights:} \qquad w_1 = w_3 = \frac{b-a}{6}, \quad w_2 = \frac{2}{3}(b-a).$$

is of order 4. An n-point Gaussian quadrature rule; that is, when $x_1, ..., x_n$ are the zeros of an nth orthogonal polynomial on $[a, b]$ with weights calculated from the Vandermonde system

$$\sum_{j=1}^{n} w_j x_j^m = \int_a^b x^m \,\mathrm{d}x, \quad m = 0, ..., n-1. \tag{6}$$

is of order $2n$.

Suppose now that we implement the quadrature rule (1) that admits an estimate of the form (5) on $[a, b]$, on a different interval $[\tilde{a}, \tilde{b}]$ where the quadrature weights and quadrature points are chosen according to (3). For a given, sufficiently smooth function $f : [\tilde{a}, \tilde{b}] \to \mathbb{R}$ let us define $g : [a, b] \to \mathbb{R}$ by

$$g(x) := f\left(\frac{b-x}{b-a}\tilde{a} + \frac{x-a}{b-a}\tilde{b}\right)\left(\frac{\tilde{b}-\tilde{a}}{b-a}\right), \quad x \in [a, b].$$

Then, according to (2) we obtain

$$\begin{aligned}\left|\int_{\tilde{a}}^{\tilde{b}} f(x)\,\mathrm{d}x - \sum_{j=1}^{n} \tilde{w}_j f(\tilde{x}_j)\right| = \left|\int_a^b g(x)\,\mathrm{d}x - \sum_{j=1}^{n} w_j g(x_j)\right| &\leq c \max_{x\in[a,b]} \left|g^{(p)}(x)\right| \\ &= c\left(\frac{\tilde{b}-\tilde{a}}{b-a}\right)^{p+1} \max_{x\in[\tilde{a},\tilde{b}]} \left|f^{(p)}(x)\right| \end{aligned} \tag{7}$$

2. Adaptive numerical integration

Let $f : [a, b] \to \mathbb{R}$ be sufficiently smooth and suppose that we have a numerical quadrature of the form (1) of order p; that is, the estimate (5) holds. Let us half the interval $[a, b]$ and apply the same quadrature rule on $[a, \frac{a+b}{2}]$ and $[\frac{a+b}{2}, b]$:

$$\int_a^{\frac{a+b}{2}} f(x)\,dx \approx \sum_{i=1}^{n} f(x_i^1)w_i^1 := Q_1$$

and

$$\int_{\frac{a+b}{2}}^{b} f(x)\,dx \approx \sum_{i=1}^{n} f(x_i^2)w_i^2 := Q_2.$$

The quadrature weights and quadrature points are chosen according to (3):

$$\begin{aligned}\tilde{x}_i^1 &= \frac{b-x_i}{b-a}a + \frac{x_i-a}{b-a}\frac{a+b}{2}, \\ \tilde{w}_i^1 &= \frac{1}{2}w_i.\end{aligned}$$

and

$$\tilde{x}_i^2 = \frac{b - x_i}{b - a}\frac{a+b}{2} + \frac{x_i - a}{b - a}b,$$
$$\tilde{w}_i^2 = \frac{1}{2}w_i.$$

According to (7) we have

$$|e_1| := \left| \int_a^{\frac{a+b}{2}} f(x)\,\mathrm{d}x - Q_1 \right| \leq c\frac{1}{2^{p+1}} \max_{x \in [a, \frac{a+b}{2}]} \left| f^{(p)}(x) \right|$$

and

$$|e_2| := \left| \int_{\frac{a+b}{2}}^{b} f(x)\,\mathrm{d}x - Q_2 \right| \leq c\frac{1}{2^{p+1}} \max_{x \in [\frac{a+b}{2}, b]} \left| f^{(p)}(x) \right|$$

Therefore, heuristically, one may expect that, if

$$\int_a^b f(x)\,\mathrm{d}x = Q + e$$

(c.f. (5)), then

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x &= \int_a^{\frac{a+b}{2}} f(x)\,\mathrm{d}x + \int_{\frac{a+b}{2}}^{b} f(x)\,\mathrm{d}x \\ &= Q_1 + e_1 + Q_2 + e_2 \approx Q_1 + \frac{e}{2^{p+1}} + Q_2 + \frac{e}{2^{p+1}} = Q_1 + Q_2 + \frac{e}{2^p}.\end{aligned}$$

Therefore, we get

$$e \approx \frac{Q_1 + Q_2 - Q}{1 - \frac{1}{2^p}}$$

and thus

$$\left| \int_a^b f(x)\,\mathrm{d}x - (Q_1 + Q_2) \right| \approx \frac{|Q_1 + Q_2 - Q|}{2^p - 1}. \tag{8}$$

Notice that the right hand side of (8) is a known (computed) quantity and thus it can be used as a heuristic measure of the error between the unknown quantity $\int_a^b f(x)\,\mathrm{d}x$ and the computed quantity $Q_1 + Q_2$. These considerations lead to the following algorithm.

(1) Fix a tolerance $\epsilon > 0$. Calculate Q, Q_1 and Q_2.
(2) Calculate $h := \frac{|Q_1+Q_2-Q|}{2^p-1}$.
(3) If $h < \epsilon$, then accept $Q_1 + Q_2$ as your approximation for $\int_a^b f(x)\,\mathrm{d}x$.
(4) Otherwise half the interval and repeat the procedure for $\int_a^{\frac{a+b}{2}} f(x)\,\mathrm{d}x$ on $[a, \frac{a+b}{2}]$ with tolerance $\frac{\epsilon}{2}$ and $\int_{\frac{a+b}{2}}^{b} f(x)\,\mathrm{d}x$ on $[\frac{a+b}{2}, b]$ with tolerance $\frac{\epsilon}{2}$.
(5) If required error has been reached in each subinterval then stop.
(6) Otherwise continue as in step (4) above on the interval(s) for which the required error tolerance has not been reached. Proceed until all quadratures have reached the required tolerance in each subinterval.

(7) Thus you end up with intervals $[a_i, b_i]$ and quadratures $Q_1^i + Q_2^i$ approximating $\int_{a_i}^{b_i} f(x)\,\mathrm{d}x$ to the required tolerance, $i = 1, ..., N$, with

$$\int_a^b f(x)\,\mathrm{d}x = \sum_{i=1}^{N} \int_{a_i}^{b_i} f(x)\,\mathrm{d}x,$$

Note that most likely you will end up with several subintervals of varying lengths. Your final approximation is then

$$\int_a^b f(x)\,\mathrm{d}x = \sum_{i=1}^{N} \int_{a_i}^{b_i} f(x)\,\mathrm{d}x \approx \sum_{k=1}^{N} (Q_1^i + Q_2^i).$$

An adaptive numerical quadrature algorithm is thus nothing else than a composite quadrature rule with subintervals that are not predetermined but obtained in an adaptive manner.

3. Tasks

We'll use the following function for testing during the lab:

$$f(x) = \frac{1}{0.01 + (x - 0.3)^2} + \frac{1}{0.04 + (x - 0.9)^2} - 6 \tag{9}$$

$$\int_0^1 f(x) = 29.858325395498671; \tag{10}$$

Exercise 1. Write a class, that has an integralEqudistance and an integralAdaptive function, which will perform the numerical integrations. The parameters of these function are the following: from (floating point), to (floating point), f (callable object), and for the Equidistant version an N which describes the number of points, and for the Adaptive version an epsilon (floating point) for the tolerance. The class should be initialized by describing the order (M) of the quadrature rule, which will be used later. The raw weights are:

- $M = 2 : [1, 1]/2$
- $M = 3 : [1, 4, 1]/6$
- $M = 4 : [1, 3, 3, 1]/8$
- $M = 5 : [7, 32, 12, 32, 7]/90$

Exercise 2. Compare the performance of the adaptive and equidistant approaches. Fix the ϵ for the adaptive method (print the number of intervals, that has been used), find approximately that N value for equidistant method, which will lead to similar error rate. Compare the running time of both method. Here the M is free choice.

Exercise 3. Write another class, with similar signature for the 2-point Gaussian quadrature rule, here you don't have to create an interface to define the order (we'll use the 2-point version).

Exercise 4. Compare the error rate of the 2-points Gaussian quadrature rule and the other rules, which order has similar error rates.

Example output:

- For order M, with $\epsilon =$ *small value*:
  - Adaptive: N steps
  - Equidistant: $>N$ steps
- Run-time for the same setup with K repeats:
  - Adaptive: t_1 sec
  - Equidistant: t_2 sec
- Gaussian adaptive error: *small value*
  - Quadrature M=2: *error*
  - Quadrature M=3: *error*
  - Quadrature M=4: *error*
  - Quadrature M=5: *error*
- Gaussian equidistant error: *small value*
  - Quadrature M=2: *error*
  - Quadrature M=3: *error*
  - Quadrature M=4: *error*
  - Quadrature M=5: *error*