Programmiertechniken

Prof. Dr. L. Pollet J. Greitemann, D. Hügel, J. Nespolo, T. Pfeffer

---

Numerical integration

Throughout this worksheet, we are going to develop some numerical integration algorithms, using C++ templates.

We summarise here the numerical algorithms that we are going to use during the tutorial.

Riemann integration (or method of rectangles). This is the simplest method, in which the domain of integration is discretised in intervals of length h and for each interval $(x, x+h)$, a rectangle of height $f(x)$ is constructed. The resulting approximation is

$$\int_a^b f(x)\,dx \approx \sum_{i=0}^{N-1} f(a+ih)\,h, \qquad \text{with} \qquad h = \frac{b-a}{N}. \tag{1}$$

The error in each interval is of order $O(h^2)$ but, upon summing over all intervals, the global error becomes of order $O(h) \equiv O(1/N)$.

Trapezoid integration (or method of trapezia). Improves on Riemann integration by considering a linear approximations between the extrema of each interval, i.e.,

$$\int_x^{x+h} f(x')\,dx' \approx \frac{f(x)+f(x+h)}{2}\,h. \tag{2}$$

Extending this result over the whole domain, one obtains

$$\int_a^b f(x)\,dx \approx \left[\frac{f(a)}{2} + \sum_{i=1}^{N-1} f(a+ih) + \frac{f(b)}{2}\right] h. \tag{3}$$

In doing this, the global error is improved by a factor of h, so that it behaves as $O(1/N^2)$.

Simpson method It is based on a quadratic approximation of the integrand, $P(x) = \alpha + \beta x + \gamma x^2$, such that $P(x \pm h) = f(x \pm h)$ and $P(x) = f(x)$. The constants α, β and γ can be readily computed by considering the simpler case $\int_{-1}^{1} f(x)\,dx = \frac{1}{3}[f(-1) + 4f(0) + f(1)]$. A generalisation of this result yields

$$\int_a^b f(x)\,dx \approx \frac{h}{3}\left[f(a) + 4f(a+h/2) + 2f(a+h) + \cdots + 4f(b-h/2) + f(b)\right] + O(h^4). \tag{4}$$

Note that the implementation of the method requires an even number of evaluations of the integrand.

1) Templates, function pointers and function objects

(a) Write an integration routine using the Simpson method, with the following prototype:

```
double simpson1(double (*integrand)(double),
                const double xmin, const double xmax, const size_t npoints);
```

The argument `double (*integrand)(double)` is a function pointer, i.e., a pointer to a function that takes one parameter of type `double` and returns a `double`, itself. In this example, we are using function pointers to pass the integrand to the integration routine.

(b) Modify the previous function into one with prototype

```
template <typename Functor>
double simpson2(Functor &integrand,
             const double xmin, const double xmax, const size_t npoints);
```

A functor is a “function class”, and it is constructed by overloading the `operator()` operator. Here is a minimal example of a functor for $f(x) = x$:

```
class Functor {
   public:
   Functor() {}
   double operator() (double x) {return x;}
};
```

(c) Further modify the integrator routine to `simpson3` by templating the type of `xmin` and `xmax`. Write a minimal working example in which these two parameters are of type `int`, and verify that this case is handled correctly. You may use the functions provided in the library `<limits>` to identify the type of the parameters at run time.

2) Performance of different integration methods

(a) In the same fashion as `simpson3` in Exercise 1(c), write the integration routine `riemann` (method of rectangles) and `trapezoid` (method of trapezia).

(b) Varying the number of intervals used in the integration, tabulate the error of the different integration methods on the following functions:

$$f(x) = const; \qquad f(x) = x; \qquad f(x) = x^2;$$

$$f(x) = \sin(x); \qquad f(x) = \sin(5x); \qquad f(x) = e^{\sin^2(x)}$$

(c) Plot the results of point 2(b) using `python` and `matplotlib`, and discuss your results.

(d) Rework the integration routines coded so far so that they can be accessed through a common interface function `integrate` that takes as extra argument the method of integration to be used.

(e) (bonus) How could you improve the error of the `riemann` integrator? Discuss the performance of the trapezoid method when integrating a periodic function over and domain that is a multiple of the period.

(f) (bonus) Why is the error of the Simpson method $O(h^4)$?

(g) (bonus) Read Ch.IV of *Numerical Recipes*.