

QUADRATURE METHODS (II)

Exercise 1 (Simple vs composite rules). So far we've seen simple quadrature rule where the integral $\int_a^b f$ is approximated by a weighted sum of the form $(b-a) \sum_{i=0}^n w_i f(x_i)$ where the w_i are obtained by using the Lagrange interpolation of f . Such an approximation is proportional to $(b-a)^n$ and $|f^{(n)}|$ (if f is smooth enough). Using large values of n is in general not a good idea (cf Runge phenomenon). To approximate integrals, it is thus better to use small values of n over multiple and small segments $[a_i, b_i]$: i.e approximate f using a piecewise polynomial interpolation : composite rules.

The simple trapezoidal rule is defined as : $I_{a,b} = (b-a)(\frac{1}{2}f(a) + \frac{1}{2}f(b))$. Let's divide $[a, b]$ into M segments with regular size $h = \frac{b-a}{M}$. Let $a_j = a + jh$, $j \in [0, M]$. The composite trapezoidal rule integrates f by applying the simple trapezoidal over each segment of size h . Find the weights w_j such that the composite trapezoidal rule can be written :

$$h \sum_{j=0}^M w_j f(a_j)$$

Exercise 2 (Legendre-Gauss quadrature : example).

Let $I(f) = \int_{-1}^1 f$, where f is a smooth function over $[-1, 1]$. Let $u_0, u_1 \in [-1, 1]$. We are interested in a rule of the form :

$$\lambda_0 f(u_0) + \lambda_1 f(u_1).$$

In this exercise, the interpolation nodes are not fixed. Find $(u_0, u_1, \lambda_0, \lambda_1)$ so as to maximize the order of the method.

Exercise 3 (Gaussian quadrature : general case).

Let $\mathcal{X} \subset \mathbb{R}$. Let w be a non-negative weight function (i.e $\int_{\mathcal{X}} w = 1$). For any polynomials P, Q defined over \mathcal{X} , their scalar product for the weight function w is defined by :

$$\langle P, Q \rangle_w = \int_{\mathcal{X}} P(x)Q(x)w(x)dx.$$

Let f be a continuous function over \mathcal{X} . We are interested in finding the optimal weights $(w_i)_i$ and nodes $(x_i)_i$ maximizing the order of the quadrature rule for the integral $I(f) = \int_{\mathcal{X}} w f$ given by :

$$\widehat{I_n(f)} = \sum_{i=0}^n w_i f(x_i)$$

Part I. Theory

Orthogonal Polynomials. Consider the orthogonal family of polynomials (P_0, P_1, \dots) obtained by applying the Gram-Schmidt process to the basis $1, X, X^2, \dots$ with the inner product $\langle \cdot, \cdot \rangle_w$ and the normalization condition $P_n(1) = 1$. First, we prove an important property of any orthogonal family of polynomials : each element P_i has i distinct simple roots in \mathcal{X} .

1. Show that P_n changes sign at least once in \mathcal{X} .
2. Let y_1, \dots, y_j the only points in \mathcal{X} where P_n changes its sign. Show that $j = n$ and conclude.

Now we move on to show that the maximum order of the method is $2n+1$ and that the nodes are unique and given by the roots of the orthogonal polynomial basis.

Necessary conditions. Assume the rule is of order $2n+1$.

3. Using a particular choice for f , show that the nodes $(x_i)_i$ must be roots of P_{n+1} .
4. Conclude the uniqueness of $(x_i)_i$ and $(w_i)_i$.

Sufficient conditions.

5. Using Euclidean division, show that the necessary conditions are sufficient for the rule to be of order at least $2n + 1$.
6. Using a specific choice of f , show that the order of the rule cannot be larger than $2n + 1$.

Examples Let $n = 1$.

- For $\mathcal{X} = [-1, 1]$ and $w(x) = \frac{1}{2}$, (P_0, P_1, \dots) are called the Legendre polynomials.
- For $\mathcal{X} = \mathbb{R}$ and $w(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}$, (P_0, P_1, \dots) are called the Hermite polynomials.
- For $\mathcal{X} = [0, +\infty]$ and $w(x) = e^{-x}$, (P_0, P_1, \dots) are called the Laguerre polynomials.
- 7. Find P_0, P_1 and P_2 . Deduce the corresponding Gauss-Legendre quadrature for $n = 1$.

Part II. Implementation In the following questions, you may use library functions referring to polynomials such as `scipy.special.roots_legendre`, `scipy.special.roots_laguerre`, etc ...

8. For $\mathcal{X} = [a, b]$, what change of variable should be applied to the nodes x_i ?
9. Implement a function `def gauss_legendre_simple(f, a, b, nodes, weights)` that approximates the integral of a function f over $[a, b]$ using the Gaussian-Quadrature rule. Test your implementation on some analytically integrable function of your choice.
10. Implement a function `def gauss_legendre(f, n, M, a, b)` that performs the composite rule of Gauss-Legendre over M sub-intervals of $[a, b]$.
11. Write an identical function but with a vectorized implementation `gauss_legendre_vectorized(f, n, M, a, b)` with numpy (does not contain any loop or list comprehensions. Hint : it should not call `gauss_legendre_simple`).
12. Visualize in one plot the integration error of $\int_2^5 \frac{1}{x}$ as a function of both n and M .
13. Write a function `def trapezoidal` that computes the composite trapezoidal rule of Ex 1. Compare the integration error with the Gauss-Legendre method with the same values of M and the number of nodes. What do you conclude?
14. **Mini-project** Implement different Gauss-quadrature rules (different set of nodes i.e different space \mathcal{X}) that approximate the probability $\mathbf{P}(X \leq a)$ for a Gaussian random variable $X \sim \mathcal{N}(0, 1)$ and some $a \in \mathbb{R}$. Evaluate the accuracy of your approximation (as a function of n) using `scipy.stats.norm.cdf`. Which rule is more accurate and why?

Part III. Richardson acceleration and Romberg's method Consider an approximation $A(h)$ of a quantity of interest A given by :

$$A(h) = A + a_1h + a_2h^2 + O(h^k)$$

The goal of Richardson's method is to eliminate the h^i terms using multiple evaluations $A(h_1), A(h_2), \dots$. Let α be a known and fixed positive constant. Setting $h_i = \alpha^i h$ for some $h > 0$, we define :

$$B(h) = \frac{A(\alpha h) - \alpha A(h)}{1 - \alpha} = A + O(h^2)$$

Thus, B eliminates the error term a_1h . This process can be repeated to eliminate the remaining terms.

14. Let $A_{i,0} = A(\alpha^i h)$ and $A_{1,1} = B(h)$. Show that the expression given by $A_{i,j} = \frac{A_{i,j-1} - \alpha^j A_{i-1,j-1}}{1 - \alpha^j}$ removes the first j error terms using i nodes.
15. Richardson's acceleration applied to the trapezoidal rule is called Romberg's integration. The error of the trapezoidal rule is given by :

$$\widehat{I_t(h)} = \int f + \sum_{k=1}^n a_k h^{2k} + O(h^{2n+2})$$

Implement Richardson's method applied to $I_t(h)$.

16. Plot the integration error as a function of $1/h$ for the trapezoidal rule with and without Richardson's acceleration. Compare with Gaussian quadrature.

Richardson acceleration can also be applied to speed-up optimization algorithms :

<https://francisbach.com/richardson-extrapolation/>