Gauss 求積実装クイックガイド

D2 佐々木　徹

May 16, 2024

対象

理論的な部分はひとまず置いて実装をしたい人向け

概要

Gauss 求積 (Gauss quadrature) の定義

積分を和の形式に表現する近似法．和にして計算コスト削減できる．

$$\int \mathrm{d}x\, f(x) \sim \sum_{i=1}^{N} w_i f(x_i)$$

積分区間および求積点 x_i と重率 w_i は多項式を選ぶと自動的に決定される．

$[a, b]$	$w(x)$	Name	Symbol	N_n
$[-1,\ 1]$	1	Legendre	$P_n(x)$	$2/(2n+1)$
$[-1,\ 1]$	$1/\sqrt{1-x^2}$	Chebyshev	$T_n(x)$	$\pi\ \ (n=m=0)$
				$\pi/2\ \ (n=m\neq 0)$
$[-1,\ 1]$	$(1-x^2)^{\lambda-\frac{1}{2}}$	Gegenbauer	$C_n^{(\lambda)}(x)$	$\frac{2^{1-2\lambda}\pi\Gamma(n+2\lambda)}{n!(n+\lambda)\Gamma^2(\lambda)}$
$[-1,\ 1]$	$(1-x)^{\alpha}(1-x)^{\beta}$	Jacobi	$P_n^{(\alpha,\beta)}(x)$	$\frac{2^{\alpha+\beta+1}}{2n+\alpha+\beta}\frac{\Gamma(n+\alpha+1)\Gamma(n+\beta+1)}{n!\Gamma(n+\alpha+\beta+1)}$
$[-1,\ 1]$	1	Sine	$\sin(n\pi x)$	1
$[-1,\ 1]$	1	Cosine	$\cos(n\pi x)$	1
$[-\infty, \infty]$	e^{-x^2}	Hermite	$H_n(x)$	$2^n\sqrt{\pi}n!$
$[-\infty, \infty]$	1	Hermite	$h_n(x) = e^{-x^2/2}H_n(x)$	$2^n\sqrt{\pi}n!$
$[0, \infty]$	$x^{\alpha}e^{-x}$	Associated Laguerre	$L_n^{(\alpha)}(x)$	$\frac{\Gamma(n+\alpha+1)}{n!}$
$[0, \infty]$	$x^{2\alpha+1}e^{-x^2}$	Sonine	$S_{\alpha}^{(n)}(x^2)$	$\frac{\Gamma(n+\alpha+1)}{2n!}$

$\Gamma(\alpha) = \int_0^{\infty} x^{(1-\alpha)}e^{-x}dx$ is the Gamma function

Python での実装

np.polynomial ライブラリ

各々の多項式における求積点 x_i と重率 w_i を与えるライブラリがある.

- Legendre 多項式: np.polynomial.legendre.leggauss
- Hermite 多項式: np.polynomial.hermite.hermgauss
- Chebyshev 多項式: np.polynomial.chebyshev.chebgauss

実装 (ex. $\int_{-1}^{+1} \mathrm{d}x\, \mathrm{e}^x$)

```python
import numpy as np
x = np.polynomial.legendre.leggauss(10)
sum_x = 0
for i in range (0, len(x[0])):
    sum_x += x[1][i]*np.exp(x[0][i])
print(sum_x)
```

結果

Gauss-Legendre quadrature, exp(x): 2.3504023872876028

Numerical integration, exp(x): 2.3504023872876028

DVR との対応

グリッド基底 $\leftrightarrow$ スペクトラル基底のユニタリー行列

スペクトラル基底として例えば，調和振動子固有関数 ($\propto$Hermite 多項式) など．固有関数基底とグリッド基底との行き来ができると便利がよい．

$$h_n(x) = \sqrt{w(x)}\frac{H_n(x)}{N_{H_n}}, \quad \int \mathrm{d}x \, h_m(x) h_n(x)$$

に対してユニタリー行列は次のように計算する．

$$\mathbb{U} = \begin{pmatrix} h_0(x_1) & h_1(x_1) & h_2(x_1) & \cdots & h_{N-1}(x_1) \\ h_0(x_2) & h_1(x_2) & h_2(x_2) & \cdots & h_{N-1}(x_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_0(x_N) & h_1(x_N) & h_2(x_N) & \cdots & h_{N-1}(x_N) \end{pmatrix}$$

DVR との対応

ユニタリー行列の実装

```
import numpy as np
import sympy as sp
x = np.polynomial.legendre.leggauss(10)
T = np.zeros((len(x[0]), len(x[0])))
for i in range (0, len(x[0])):
    for j in range (0, 2*dimension + 1):
        wj = x_legendre[1][j]
        Ni = 2/(2*i + 1)
        T[i][j] = float(np.sqrt(wj)*np.sqrt(1/Ni)
            *sp.legendre(i, x[0][j]))
print(T)
```