Norwegian University of Science and Technology
Institutt for matematiske fag

TMA4130/35 Mathematics 4N/4D Høst 2021

Exercise set 2

Submission deadline: **Tuesday, Sep 21 2021 at 12:00 (noon)**

In this exercise set we will construct and analyze quadrature rules. For guidance on quadrature rules, please read the lecture notes. Make sure to run the code below to get all the important modules.

[]:
```
%matplotlib inline

from numpy import *
from matplotlib.pyplot import *
from math import factorial
import matplotlib.pyplot as plt

newparams = {'figure.figsize': (8.0, 4.0), 'axes.grid': True,
             'lines.markersize': 8, 'lines.linewidth': 2,
             'font.size': 14}
plt.rcParams.update(newparams)
```

1 Analyzing the composite Simpson's rule

Simpson's rule is defined as

$$S[f](x_{i-1}, x_i) = \frac{h}{6}(f(x_{i-1} + 4f(x_{i-1/2}) + f(x_i))$$

where $h = x_i - x_{i-1}$ and $x_{i-1/2} = \frac{x_{i-1}+x_i}{2}$.

a) Show that the resulting composite Simpson's rule is given by

$$\int_a^b f\,\mathrm{d}x \approx \mathrm{CSR}[f]([x_{i-1}, x_i]_{i=1}^m) = \tfrac{h}{6}[f(x_0) + 4f(x_{x_{1/2}}) + 2f(x_1) + 4f(x_{3/2}) + 2f(x_2) + \ldots + 2f(x_{m-1}) + 4f(x_{x_{m-1/2}}) + f(x_m)].$$

b) Implement the composite Simpson's rule. Use this function to compute an approximated value of the integral

$$I(0,1) = \int_0^1 \tan\left(\frac{\pi}{4}x\right) = 2\frac{\log(2)}{\pi} = 0.4412712\ldots.$$

for $m = 4, 8, 16, 32, 64$ corresponding to \$ h = 2^{-2}, 2^{-3}, 2^{-4}, 2^{-5}, 2^{-6}\$. Tabulate the corresponding quadrature errors $|I(0,1) - \mathrm{CSR}[f]([x_{i-1}, x_i]_{i=1}^m)|$. Plot the errors against h in a log − log plot and determine the EOC ("Experimental Order of Convergence") How does it compare to the composite trapezoidal rule?

c) Recall that the error of Simpson's rule on a single interval is given by

$$|I[f](a,b) - S[f](a,b)| = -\frac{(b-a)^5}{2880} f^{(4)}(\xi)$$

for some $\xi \in [a,b]$.

Use this to show that the error of the composite Simpson rule can be bounded by

$$|I[f] - \mathrm{CSR}[f]| \leqslant \frac{M_4}{2880}\frac{(b-a)^5}{m^4} = \frac{M_4}{2880} h^4 (b-a) \tag{3}$$

where $M_4 = \max_{\xi\in[a,b]} |f^{(4)}(\xi)|$. Does your numerical experiments from b) support the theoretically derived convergence order?

d) Redo the numerical experiment from b), but this time, use the composite Simpson rule to compute approximated values of the integral

$$\int_0^1 \sqrt{1-x^2}dx = \frac{\pi}{4}.$$

What EOC do you obtain? Do you have an explanation for reduced convergence order?

Hint. Have a look a f', e.g. by plotting f' over intervals $[0,b]$ with $0 < b < 1$ but b very close to 1, e.g. $b = 0.9999$

2 Gaussian Quadrature

In this exercise we will construct a Gaussian quadrature rule with 3 nodes. We will take it step by step, so don't worry if you do not feel like an expert on Gaussian quadrature.

To make your life easy, we will use the sympy python module for symbolic mathematics to perform tasks such as (symbolic) integration and root finding of low order polynoms. In particular look at integrate and solve submodules.

The first step in constructing a Gaussian quadrature is finding the correct orthogonal polynomial. The nodes of the quadrature rule will be the roots of some polynomial. Since we are looking for 3 nodes, this means that the polynomial should have 3 roots, and hence we are looking for a third-order polynomial.

The polynomial, call it p_3, should be orthogonal on the interval $[-2,1]$ to all polynomials of order 2 or less. We now create this polynomial.

Start with the 4 polynomial "basis" functions

$$\phi_0 = 1, \quad \phi_1 = x, \quad \phi_2 = x^2, \quad \phi_3 = x^3.$$

Remember that on the interval $[0,3]$ we have the *inner product*

$$\langle p, q\rangle = \int_{-2}^{1} p(x)\, q(x)\, \mathrm{d}x$$

and the *norm*

$$\|p\| = \left(\int_{-2}^{1} p(x)^2\, \mathrm{d}x\right)^{1/2}.$$

We can now construct orthogonal polynomials by using Gram-Schmidt orthogonalization.

$$p_k = \phi_k - \sum_{j=0}^{k-1} \frac{\langle \phi_k, p_j \rangle}{\|p_j\|^2} p_j$$

We start out by setting $p_0 = 1$. In order to calculate p_1 we first need to calculate

$$\langle \phi_1, p_0 \rangle = \int_{-2}^{1} \phi_1(x)\, p_0(x)\, \mathrm{d}x = \int_{-2}^{1} x \cdot 1 \, \mathrm{d}x = \left[\frac{x^2}{2}\right]_{-2}^{1} = -\frac{3}{2}.$$

We also need to calculate

$$\|p_0\|^2 = \int_{-2}^{1} p_0(x)^2 \, \mathrm{d}x = \int_{-2}^{1} 1 \cdot 1 \mathrm{d}x = 3.$$

Therefore,

$$p_1 = \phi_1 - \frac{\langle \phi_1, p_0 \rangle}{\|p_0\|^2} p_0 = \phi_1 - \frac{-3/2}{3} \cdot p_0 = \phi_1 + \frac{1}{2} p_0 = x + \frac{1}{2}.$$

We can use the Python package SymPy to check our calculations. The code below helps you by defining the inner product and shows how to define polynomials.

[]:
```
from sympy.abc import x
from sympy import integrate

a=-2
b=1

#Define the inner poduct
def scp(p,q):
    return integrate(p*q, (x, a, b))

#Define polynomials
p0 = 1
phi1 = x

#Calculate the inner product and print it.
print(scp(p0,phi1))
```

```
-3/2
```

a) Use Gram-Schmidt orthogonalization to construct p_2 and p_3.

b) Use the function `scp` to check whether the polynomials you calculated are in fact **orthogonal**.

c) Find the 3 roots of p_3.

- Analytical approach: If you want to do it analytical, use the fact that one root is

$$x = -\frac{1}{2}$$

to find a second order polynomial $\widetilde{p}_2$ such that $\widetilde{p}_2(x) \cdot (x + 1/2) = p_3(x)$.

- Computational approach: If you want to use a computational algebra system/symbolic calculator you Import the `solve` from `sympy` (Have a look at the solve submodules.)

d) Let's denote the three roots of p_3 by x_1, x_2, x_3.

Construct the three Lagrange polynomials l_1, l_2, l_3 satisfying $l_i(x_j) = \delta_{ij}$, that is

$$l_i(x_j) = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

Then calculate the weights

$$w_i = \int_{-2}^{1} l_i(x)\,\mathrm{d}x.$$

Hint: You can use the SymPy function `integrate` *to check your calculations.*

e) Now recheck your calculations as follows. The Gauss-Legendre rule for the interval $[-1, 1]$ with 3 quadrature points is given by

$$\{\hat{x}_i\}_{i=0}^{2} = \left\{-\sqrt{\tfrac{3}{5}}, 0, \sqrt{\tfrac{3}{5}}\right\}$$
$$\{\hat{w}_i\}_{i=0}^{2} = \left\{\tfrac{5}{9}, \tfrac{8}{9}, \tfrac{5}{9}\right\}$$

Now transfer this quadrature rule to the interval $[0, 3]$ and confirm that you get the same quadrature points and weights you computed in 2a)-2d).

f) Finally, write down the quadrature rule on the form

$$\mathrm{GQR}[f](0,3) = \sum_{j=1}^{n} w_j f(x_j).$$

and check that this Gaussian quadrature rule has degree of exactness equal to 5.

Hint: Use the `QR` *function from the* `SimpleQuadrature.ipynb` *notebook.*

3 More on the Gauss-Legendre quadrature with 3 quadrature points

a) The error of Gauss-Legendre quadrature you developed in problem 2) over one interval is given by

$$E(a,b) = \int_a^b f(x)dx - Q(a,b) = \frac{(b-a)7}{2016000} f^{(6)}(\eta), \quad \eta \in (a,b).$$

Use this to find an error expression for the composite Gauss-Legendre quadrature

$$Q_m(a,b) = \sum_{i=0}^{m-1} Q(x_i, x_{i+1}),$$

where $Q(x_i, x_{i+1})$ is the basic quadrature over the interval $[x_i, x_{i+1}]$, $x_i = a + ih$, $i = 0, 1, \ldots, m$ and $h = (b-a)/m$.

b) Based on the error expressions for the Composite Gauss-Legendre derived in 3a), find the number of intervals m that guarantee that the error in the respective composite quadrature methods is less than 10^{-8} when applied to $\int_0^1 \cos(\pi x/2)dx$.

Hint: use that $\left|\frac{d^6}{dx^6}\cos \pi x/2\right| \leq 17$ for $x \in [0,1]$.

c) Redo the previous exercise 3b) with the composite Simpson's rule instead where you can use that $\left|\frac{d^4}{dx^4}\cos \pi x/2\right| \leq 6.1$ for $x \in [0,1]$. Which composite method would you prefer if want to compute $\int_0^1 \cos(\pi x/2)dx$ numerically?

d) Based on $Q_1(a,b)$ and $Q_2(a,b)$, find an error estimate for $Q_2(a,b)$ as an approximation to $\int_0^1 \cos(\pi x/2)$.

Hint: Again, you use that $|f^{(6)}(x)| < 17$ over the interval $[0,1]$.

e) Write a function implementing the Gauss-Legendre quadrature with error estimate from d). Compare the exact error and the error estimate.