1. Assignment 2

In assignment 1, those reductions involving edge functions leave us a problem: HOW TO DO THE NUMERICAL INTEGRATION? In this assignment, you will be asked to handle this problem, and we restrict ourselves to 1D.

Recall that we need to discretize

$$f(\xi) = \pi \cos(\pi\xi) \ ,$$

with edge functions

$$e_i(\xi), \quad i = 1, 2, \cdots, N,$$

into

$$f^h(\xi) := \sum_{i=1}^{N} f_i e_i(\xi) \ ,$$

where $f_i = \int_{\xi_{i-1}}^{\xi_i} f(\xi)\mathrm{d}\xi$. This process of obtaining f_i from $f(\xi)$ is called **reduction**.

If we can get the analytical expression of this integral, there will be no problem at all. But that is not always the case. Therefore, we need to have our numerical integration module.

Although, with Python, it is possible to do the reduction for all line segments $[\xi_{i-1}, \xi_i]$ at once high efficiently, it, as this is your first try, will be very okay if you do it in a *for* loop where the following module is called repeatedly:

```
def numerical_integration_1d(func, quad_degree, quad_type='gauss', a=-1, b=1):
    """
    This module do numerical integration of 'func' over domain [a, b] using
    'quad_type' quadrature at degree 'quad_degree'.
    """
    ......
```

The algorithm has been mentioned during our meeting and in your Applied Numerical Analysis course I think.

Next, we need to compute the L^2-error of the reconstructed data. It will not be a big problem if you have done the numerical integration module. Al last, please play around with different polynomial degrees and different functions, and then do some p-convergence plots to see if exponential or algebraic convergence is obtained.