

Series 0



Computational Methods for
Engineering Applications
Last edited: September 14, 2020
Due date: No due date at 23:59

Template codes are available on the course's webpage at <https://moodle-app2.let.ethz.ch/course/view.php?id=13412>.

This is a warmup problem. You do not need to hand in this problem.

Exercise 1 Midpoint rule

In this exercise we let $a < b$ be two real numbers and $f : [a, b] \rightarrow \mathbb{R}$ be a smooth function. Our goal is to approximate the integral

$$I(f) := \int_a^b f(x) \, dx.$$

Recall that for a given number of subintervals n , the *midpoint rule* $I_n(f)$ is given as

$$I_n(f) := \frac{b-a}{n} \left[\sum_{k=0}^{n-1} f\left(a + (k+1/2) \frac{b-a}{n}\right) \right].$$

It can be checked that the error scales as $\mathcal{O}(n^{-2})$, in other words

$$|I(f) - I_n(f)| \leq Cn^{-2}.$$

1a)

Write a function in C++ that computes and returns (as `double`) the midpoint rule. Use the following signature

```

#pragma once
///
/// This is the type of a function taking as parameter a double, and
/// return a double
///
typedef double(*FunctionPointer)(double);

///
/// Computes the midpoint rule to approximate the integral
///
/// \param a the left endpoint
/// \param b the right endpoint
/// \param n the number of subintervals to use
/// \param f the function to compute the integral over
///
double midpoint_rule(double a, double b, int n, FunctionPointer f);

```

1b)

For the rest of the problem, we set

$$a = 0.2, b = 1.3, \text{ and } f(x) = \sin(\pi x).$$

Compute the exact integral $I(f)$.

1c)

Write a C++ program that computes and prints $I_n(f)$ for $n = 100$. You may use the template found in `midpoint/test_single/test_single.cpp`.

1d)

In this exercise we will investigate the experimental order of convergence for the midpoint rule. Write a C++ program that computes the difference

$$|I(f) - I_n(f)|,$$

for

$$n = 2^k \quad k = 4, 5, \dots, 11.$$

Store the output to file and plot the results in MATLAB/Python using log scales on both axes. How does this plot agree with the error bound

$$|I(f) - I_n(f)| \leq Cn^{-2}?$$

See `midpoint/test_convergence/test_convergence.cpp` for a template.

Exercise 2 Linear regression

In order to detect heart diseases in cats, a biologist asks us to predict the weight of cats' hearts (\mathbf{Y}) with their body weight (\mathbf{X}). We consider the following data ¹

Body weight (kg)	2	2.2	2.4	2.2	2.6	2.2	2.4	2.4	2.5	2.7	2.6	2.2	2.5	2.5	2.5
Heart weight (g)	6.5	7.2	7.3	7.6	7.7	7.9	7.9	7.9	7.9	8.0	8.3	8.5	8.6	8.8	8.8

and propose the next linear regression

$$\mathbf{Y} = \beta_1 \mathbf{X} + \beta_0, \quad (1)$$

where $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{15}$ are the (column) vectors containing the cats' body and heart weights, respectively.

2a)

Use the **Eigen** Library to write a **C++** code that finds the coefficients β_0 and β_1 by solving the least square problem:

$$\min_{\beta \in \mathbb{R}^2} \|\mathbf{Y} - \mathbf{A}\beta\|, \quad (2)$$

with $\beta = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$, and $\mathbf{A} = \begin{pmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_{15} \end{pmatrix}$.

Hint: Remember from your linear algebra lecture that this boils down to solve the associated normal equation $\mathbf{A}^T \mathbf{A} \beta = \mathbf{A}^T \mathbf{Y}$.

Hint: The **Eigen** LU solver might be of use.

¹adapted from the dataset `cats` in R.