GCH8108E
HW3 - Interpolation and the Finite Element Method
January 30, 2025

This homework consists of two problems and will count towards 10% of your total grade for the semester. You need to hand in your code (in a single .zip file and without build files) and a LaTeX report describing the solution of each problem (use the template provided). **Submission deadline:** Thursday, February 13th before the class (13h45).

1 Interpolating using Lagrange Polynomials

In this question, we will acquire hands on experience with using Lagrange Polynomials to interpolate a function which is known *a priori*. To do so, familiarize yourself with the following two classes of the library, `FEValues<dim>` and `FE_Q<dim>`. These are the essential objects to interpolate over cells. We will also need to familiarize ourselves with the notion of the `DoFHandler<dim>` class and the `Vector<double>` class which will be used respectively to allocate the degrees of freedom associated with an interpolation support and to store the data.

Follow the instructions in the `interpolation.cc` code, finish the `setup_triangulation()` and `calculate_L2_error()` functions to answer the following questions:

- Interpolate the function $\sin(3x)$ for $x \in [0, 10]$ using first-order (Q1), second-order (Q2) and third-order (Q3) Lagrange Polynomials assuming the domain is divided into 4, 8, 16 and 32 cells. Plot the result of the interpolation and the analytical function on the same graphic.

- Fill the `calculate_L2_error()` function of the class and use it to evaluate the influence of the mesh and of the order on the accuracy of the interpolation. Discuss the influence of the order of interpolation and the number of subdivisions of the domain on the accuracy of the interpolation.

As you will see in the `interpolation.cc` code, the interpolation order and the initial refinement level is passed as a constructor argument to the main class of the solver. This will enable you to reuse the same code to solve all parts of the problem.

2 Solving the heat equation using the Finite Element Method

After all this time, we finally venture into solving a PDE using the Finite Element Method with `deal.II`. In this problem we will solve 1D, 2D and 3D problems. The class you will create will be generic and is located in the `heat_equation.cc` code. The same code (albeit with small ifs there and there), will be used to solve the heat equation.

2.1 1D problem

The first problem is the 1D heat equation:

$$\frac{\mathrm{d}^2 T}{\mathrm{d}x^2} = 1 \tag{1}$$

on a domain Ω such that $x \in [0, 5]$ and with Dirichlet boundary conditions such that $T(x = 0) = 0$ and $T(x = 5) = 0$. Complete the code to solve this problem.

(a) Find the analytical solution to this problem;

(b) Using the completed code, illustrate the solution that you obtain using Q_1 elements for a few mesh;

(c) Using the completed code, illustrate the solution you obtain using Q_2 elements. What do you observe?

2.2 2D problem

The second problem is the 2D heat equation:

$$\nabla^2 T = 0 \tag{2}$$

on a domain Ω which is a 2D hyper shell with an inner radius of $R_i = 0.25$ and outer radius $R_O = 1$. The problem is solved with two Dirichlet boundary conditions such that $T(r = R_i) = 0$ and $T(r = R_O) = 1$. We solve this problem in Cartesian coordinates.

(a) Find the analytical solution to this problem in cylindrical coordinates;

(b) Using the completed code, illustrate the solution that you obtain using Q_1 elements for a few meshes;

(c) By calculating the $\mathcal{L}^2$ norm of the error, show that the solver preserves the order of convergence of the FEM scheme.

2.3 3D problem

Define a problem of your choice in 3D. You can use a geometry provided by the GridGenerator.

(a) Define the problem completely (equations, boundary conditions and physics);

(b) Illustrate the solution you obtain in an adequate manner and analyze the results obtained.