

Assignment 1

Shushu Qin¹

¹*Laboratori de Càlcul Numèric (LaCàN), ETS de Ingenieros de Caminos, Canales y Puertos, Universitat Politècnica de Catalunya, Barcelona, Spain*

March 14, 2019

This report is the second part of the implementation of a Finite Element Program. Following the design of the program submitted in Assignment 1, we have divided the report into two sections. This first section summarises the key ideas of the program, then followed by the description of the functions designed previously with some comments on the implementation followed in Matlab. On the other hand, the last section is a summary of functions that were not considered at the start but were needed to successfully implement the FEM program.

1 Implementation Finite Element Method for 2D Laplace equation

1.1 Main description of the code

Input values entered by the user : Domain, Neumann BC ($\partial\Omega_N$), Dirichlet BC ($\partial\Omega_D$), material property μ , source term S, the number of nodes in the x and y directions (nx; ny), the type of elements (tri,quad and linear,quadratic).

Change
this

All those functions are contained into **mainFEM2D**, it calls the smaller functions in the order described in the 'Structure of the code' section.

1.2 Description of the functions associated to the code

★ PREPROCESSING

ReferenceElement

- **Inputs:** mesh (**X** and **T**)
- **Outputs:** Reference element data (shape functions, derivatives, locations and weights of the Gauss points)
- **Description:** Creates the reference element that will be used throughout the program. The reference element is defined with respect to the isoparametric variables ξ and η .
- **Uses:** None
- **Used by:** Function *BuildSystem*.
- **Comments:**
 - ★ Based on the information of the mesh, this function computes and returns the reference element data.

Comments on the implementation

- ★ The input parameters considered on the design of the program are not the same as the ones used on the implementation. We considered that using the type and degree of the elements was a better option when computing the reference element for the problem.
- ★ For each type, we define the number of nodes (*nen*) and Gauss points in each element(*ngaus*).
- ★ The function *ReferenceElement* calls two new function that respectively calculate the Gauss integration points and weights and the shape functions and their derivatives. They are called **Quadrature** and **ShapeFunc**.
- ★ The data obtained in this function is stored as a data structure called **Element**. It will be called throughout the code using the format *Example.value* depending on the information that we seek.