

PROGRAMMING FOR ENGINEERING AND SCIENTISTS

Homework 2a: Design of a FE program using C++

Author:

Luis Ángel AVILÉS MURCIA

luis.angel.aviles@upc.edu

Mariano Tomás FERNANDEZ

m.tomasfernandez@gmail.com

Shardool KULKARNI

shardoolkulkarni1996@gmail.com

Professors:

Marco DE CORATO

Sergio ZLOTNIK

June 10rd, 2020
Academic Year 2019-2020

Contents

1	Introduction	1
2	Design of the Code	1
2.1	Possibilities of the FEM code	1
2.2	Code structure	1
2.2.1	User inputs	1
2.3	Classes	1
2.3.1	FEMObject	2
2.3.2	ProblemInputs	2
2.3.3	Mesh	2
2.3.4	Element	2
2.3.5	Gauss Class	2
2.3.6	GlobalMatrix	2
2.3.7	FEMSolver	2
2.4	Post Processing	4
3	Comments on the design	5

1 Introduction

This assignment requires the design of a FEM code to solve the Poisson's problem in the programming language C++. The first part address an initial design of the FE code. This part involves no coding. The FE code has to be as flexible as possible in order to deal with different geometries, boundary conditions, loads and materials among others. This part of the assignment involves no coding but general descriptions of each of the component (classes, objetos, methods, etc) typical of the Object Orient Programming using C++.

2 Design of the Code

Different to the procedural approach used in other programming language like Matlab or Fortran, C++ is an abstract data type which works with objects and classes capable of performing predefined actions. Following we present the scope of our FEM code (according with the requirements from the assignment) and the way how to deal with each part.

2.1 Possibilities of the FEM code

At least, the following possibilities should be considered in our code:

- 2D/3D domains
- Triangular/Tetrahedral elements
- Dirichlet/Neumann boundary conditions
- Material properties depending on space location
- Initial conditions

2.2 Code structure

The code consists of classes and objects that will communicate through "messages" to perform the calculations, assemblies and solving the system of equations that typically arise in a FEM code. The code scheme is described in Figure 1.

2.2.1 User inputs

These shall include at least the following:

- Geometry;
- Boundary conditions;
- External forces/heat sources;
- Materials;
- Initial conditions (depending on the nature of the problem);
- Element type: triangular, quadrilateral
- Interpolation type: linear, quadratic
- Finally, the user should check that the mesh to determine if the results will be good enough.

2.3 Classes

There are some basics classes that a FEM code need to be implemented in a object oriented programming. The purpose of this classes is to be simple and descriptive, as well as, intuitive. Following we present the classes that we need basic for our implementation in order to solve the Poisson equation.

2.3.1 FEMObject

This class is inherited from the execution of the program and is the main responsible for the method to be executed. This class will require inputs and give results, Figure 2.

2.3.2 ProblemInputs

This class will store the particularities of the problem such as domain, material properties, loads and boundary conditions. Is inherited from the `FEMObject` class and inherited by `Mesh`, see Figure 2.

2.3.3 Mesh

This function gets the description of the domain and generates a mesh and connectivity matrix. Inherits from `ProblemInputs` where the interpolation degree is defined and is inherited.

2.3.4 Element

Elements will be considered as a structure variable which contains several characteristics of the element, for instance material, Gauss points, shape functions, etc. This structure form allows to collect all properties of a variable or object and help having a clearer code. The properties of the object will be name as; `element.shapeFunction` to indicate the shape function of the element, `element.gaussPoint` to indicate the gaussian point of the element and so on.

2.3.5 Gauss Class

Contains the methods to get the constructor of Gauss points positions and weight of a reference element that is used to locate the integration points of the whole domain.

2.3.6 GlobalMatrix

This class constructs the assembler of the element stiffness and load vector. The method use all the attributes from the elements as well as the output from the other classes.

2.3.7 FEMSolver

This class create the constructor to implement the method for solving the system of equations who arise from the assembly object. There are many methods for solving typical system of equation in FEM, but we will use the Lagrange multipliers method.

The figure 1 show the diagram of the designed classes and next figures show the inheritances, attributes and methods of each class defined.

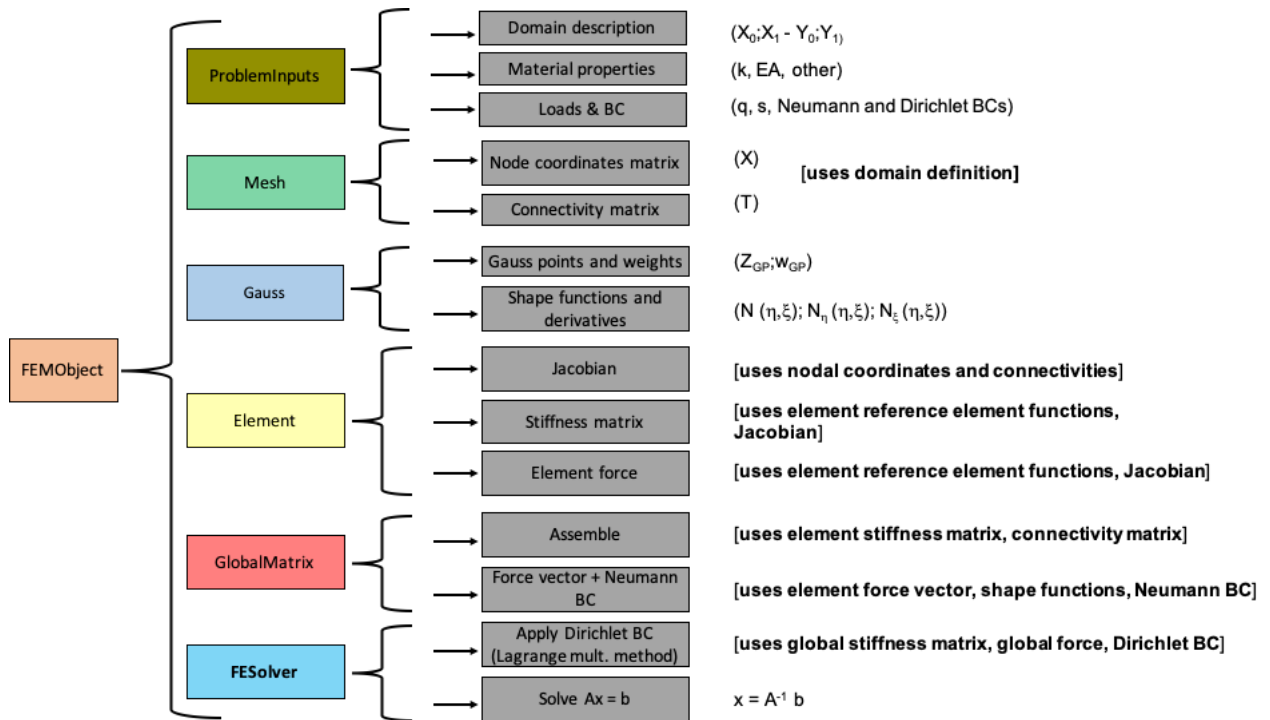


Figure 1: General flow chart of problem solution

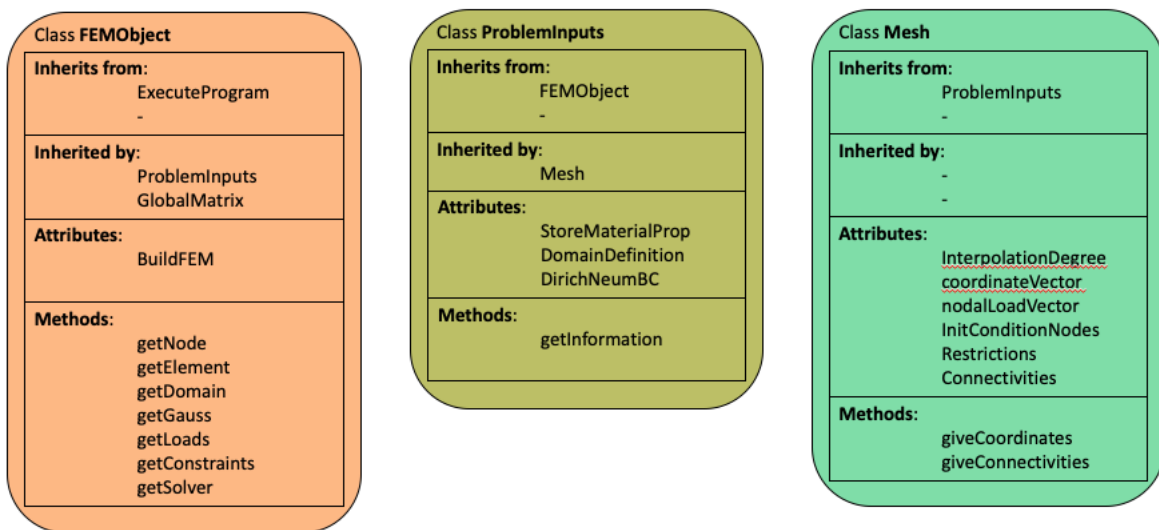


Figure 2: FEMObject, ProblemInputs and Mesh classes description

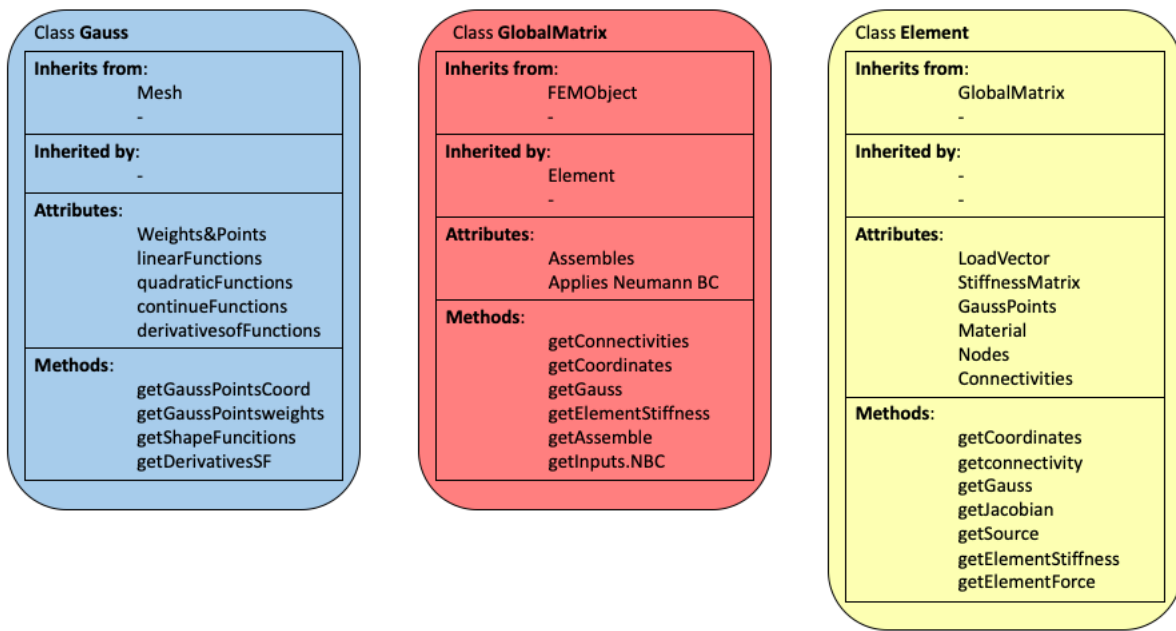


Figure 3: Gauss, GlobalMatrix and Element classes description

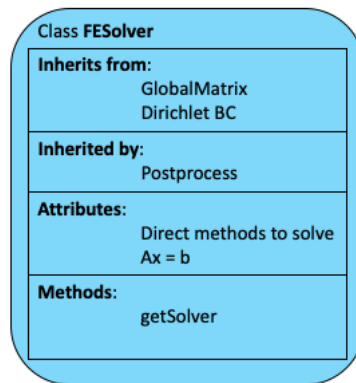


Figure 4: FESolver class description

2.4 Post Processing

The output file, along with nodes coordinates and connectivities can be plotted using either matplotlib library of Python or MATLAB itself. The data files can be imported and the post processed to generate the plots and figures. We can also create vtk files with the help of MATLAB as we have done in the previous assignment to view the results in Paraview.

3 Comments on the design

Some additional features of a FEM code includes:

- This code structure follow the GiD structure of pre-procesing like material definition, boundary conditions specifications and external forces applied.
- The user has to define use Lagrange or Serendipity rectangular mesh in Mesh Menu in GiD – Quadratic Type option.
- Since C++ is a lower level language than MATLAB which we have used previously it handles larger matrix systems faster.
- For most of the simulation problems, a quadratic element is enough to get accuracy (with a minimum error) in the results. For this reason, we consider that linear and quadratic element type are enough to run a problem and we will no consider cubic or quartic elements.
- Eigen Linear Algebra library is used to define and build matrices and vectors, and finally to solve the final system of equations.