# Finite Element Analysis (FEA) Application – Functional Specification Document

Version: 1.0  
Date: 2025-03-07  
Author: Muhanned Samir Yacoub Ma’ayeh

# 1. Introduction

## 1.1 Purpose

This document provides a detailed functional specification for the Finite Element Analysis (FEA) application. The application is designed for educational and experimental purposes, allowing users to define, simulate, and visualize basic finite element problems.

## 1.2 Scope

The system will provide:  
- A graphical user interface (GUI) for defining FEA models.  
- Mesh generation and element assignment capabilities.  
- Material and cross-section lookup tables stored in HDF5.  
- A solver engine for computing displacements and stresses.  
- Post-processing features for visualization.  
- Data persistence through HDF5 storage.

# 2. System Overview

The FEA application consists of:  
- A user interface for defining problems and displaying results.  
- A computational core that handles matrix assembly and solving.  
- A database system based on HDF5 for storing input and output data.  
- A post-processing module for visualization of results.

# 3. Functional Requirements

This section outlines the key functional requirements of the system.

## 3.1 User Interface (GUI)

The GUI shall allow users to:  
- Define geometry, mesh, materials, and boundary conditions.  
- Run the FEA solver and visualize results.  
- Save and load simulation data from HDF5 files.

## 3.2 Mesh Handling

The system shall support:  
- Importing mesh data from external files.  
- Generating structured and unstructured meshes.  
- Assigning elements to predefined material properties.

## 3.3 Material & Cross-Section Database

The system shall:  
- Store predefined material properties (Steel, Aluminum, Concrete, etc.).  
- Store predefined cross-section properties (I-beam, Rectangular, Circular).  
- Allow users to define custom materials and cross-sections.

## 3.4 Solver Engine

The solver shall:  
- Assemble the global stiffness matrix based on defined elements.  
- Apply user-defined loads and boundary conditions.  
- Solve for nodal displacements using direct or iterative numerical methods.

# 4. User Interface Design

The GUI shall include the following components:  
- \*\*Mesh Viewer\*\*: Display 2D/3D finite element meshes.  
- \*\*Material Assignment Panel\*\*: Assign materials to elements.  
- \*\*Solver Control Panel\*\*: Start, pause, or stop simulations.  
- \*\*Results Viewer\*\*: Display displacements, stresses, and strains.

# 5. Data Storage & Formats

All simulation data will be stored in an HDF5 database. The data structure includes:

HDF5 Structure:  
```  
fea\_data.h5  
├── /inputs  
│ ├── nodes [dataset]  
│ ├── elements [dataset]  
│ ├── materials [group]  
│ ├── boundary\_conditions [dataset]  
├── /results  
│ ├── displacements [dataset]  
│ ├── stresses [dataset]  
│ ├── strains [dataset]  
└── /metadata  
 ├── timestamp [attribute]  
 ├── solver\_version [attribute]  
```

# 6. Algorithms & Solver Specifications

The FEA solver will implement:  
- \*\*Finite Element Method (FEM)\*\* for structural analysis.  
- \*\*Numerical Integration\*\* using Gaussian quadrature.  
- \*\*Direct Solvers\*\* (e.g., LU decomposition) for small problems.  
- \*\*Iterative Solvers\*\* (e.g., Conjugate Gradient) for large systems.

# 7. Constraints & Assumptions

The system assumes:  
- Users will provide well-defined meshes and inputs.  
- Only small to medium-sized problems (up to 100,000 nodes) will be solved.  
- No advanced non-linear or transient analysis is included.

# 8. Performance & Security Considerations

The system shall:  
- Optimize memory usage for large simulations.  
- Prevent unauthorized modification of predefined databases.  
- Include a disclaimer to warn against professional use.