

AutoFEA-ML (Smart FEA + AI) Project Report

1. Introduction

This project demonstrates the integration of Python-based Finite Element Analysis (FEA) with an AI-assisted workflow, using a React frontend for visualization. The aim is to simplify pre-processing, simulation, and post-processing of mechanical structures using an automated pipeline.

2. Project Structure

- Backend (FastAPI + Python): Handles data input, FEA computation, and returns JSON results.
- Frontend (React + Vite): Provides an interactive interface for entering input parameters and visualizing results.
- Docs: Contains demonstration script and report for project documentation.
- Data: Sample input files and results storage (optional).

3. Methodology

- Users provide input parameters such as load and cross-sectional area.
- The backend calculates:
 - Node displacements (u)
 - Element forces
 - Element stresses
 - Maximum stress and displacement
- Results are returned in JSON format to the frontend.
- The frontend visualizes the data through charts, tables, and optional plots for better understanding.

4. Sample Results

For a test input: Load = -1000 N, Cross-sectional area $A = 0.0001 \text{ m}^2$, the backend output includes:

- Displacements: [0.0, 0.0, -4.76e-5, ...]
- Element forces: [-1000.0, -1000.0, -1414.21, ...]
- Element stresses: [-1e7, -1e7, -1.41e7, ...]

- Maximum stress: 1.41×10^7 Pa
- Maximum displacement: 0.00036 m

5. Conclusion

This project provides a streamlined workflow to integrate mechanical simulations with AI-assisted automation. Users can easily input parameters, run simulations, and visualize results without manual scripting in Abaqus or Ansys. The system is extensible, allowing future integration with other FEA solvers, additional AI algorithms, or enhanced visualization components.

6. Future Work

- Add 3D visualization of deformations
- Implement automatic report generation for all simulations
- Integrate with cloud computing for large-scale FEA
- Use AI to predict optimal structural parameters before simulation

گزارش پروژه (AutoFEA-ML (Smart FEA + AI

۱. مقدمه

این پروژه نشان می‌دهد چگونه می‌توان تحلیل اجزای محدود (FEA) مبتنی بر پایتون را با یک جریان پردازش هوش مصنوعی ترکیب کرد و خروجی آن را از طریق یک رابط کاربری تحت وب (React) به نمایش گذاشت. هدف اصلی، ساده‌سازی مراحل پیش‌پردازش، شبیه‌سازی و پس‌پردازش سازه‌های مکانیکی در قالب یک زنجیره پردازش خودکار است.

۲. ساختار پروژه

- **بک‌اند: (FastAPI + Python)** مدیریت ورودی داده‌ها، انجام محاسبات FEA و بازگرداندن نتایج به صورت JSON.
- **فرانت‌اند: (React + Vite)** فراهم کردن رابط کاربری تعاملی برای ورود پارامترها و نمایش نتایج.
- **Docs** : شامل اسکرپت‌های نمایشی و مستندات پروژه.
- **Data** : شامل فایل‌های ورودی نمونه و محل ذخیره نتایج (اختیاری).

۳. روش اجرا

- کاربر پارامترهایی مانند بار و سطح مقطع را وارد می‌کند.
- بک‌اند محاسبه می‌کند:

○ جابجایی گره‌ها (u)

○ نیروهای المان‌ها

○ تنش‌های المان‌ها

○ حداکثر تنش و جابجایی

- نتایج به صورت JSON به فرانت‌اند ارسال می‌شوند.

- فرانت‌اند داده‌ها را در قالب نمودار، جدول و ترسیم‌های اختیاری برای درک بهتر نمایش می‌دهد.

۴. نتایج نمونه

برای ورودی آزمایشی: بار -1000 N ، سطح مقطع $A = 0.0001\text{ m}^2$ ، خروجی بک‌اند شامل موارد زیر است:

• جابجایی‌ها: $[0.0, 0.0, -4.76\text{e-}5, \dots]$:

• نیروهای المان‌ها: $[-1000.0, -1000.0, -1414.21, \dots]$:

• تنش‌های المان‌ها: $[-1\text{e}7, -1\text{e}7, -1.41\text{e}7, \dots]$:

• حداکثر تنش: $1.41\text{e}7\text{ Pa}$:

• حداکثر جابجایی: 0.00036 m :

۵. نتیجه‌گیری

این پروژه یک جریان کاری ساده و کارآمد برای ترکیب شبیه‌سازی‌های مکانیکی با اتوماسیون مبتنی بر هوش مصنوعی ارائه می‌دهد. کاربران می‌توانند به راحتی پارامترها را وارد کرده، شبیه‌سازی انجام دهند و نتایج را بدون نیاز به اسکریپت‌نویسی دستی در Abaqus یا Ansys مشاهده کنند. این سیستم قابلیت توسعه داشته و می‌تواند در آینده با حل‌گرهای FEA دیگر، الگوریتم‌های هوش مصنوعی جدید یا اجزای تصویری پیشرفته‌تر یکپارچه شود.

۶. کارهای آینده

- افزودن نمایش سه‌بعدی تغییر شکل‌ها
- پیاده‌سازی تولید خودکار گزارش برای همه شبیه‌سازی‌ها
- اتصال به رایانش ابری برای اجرای تحلیل‌های FEA در مقیاس بزرگ
- استفاده از هوش مصنوعی برای پیش‌بینی پارامترهای بهینه سازه قبل از شبیه‌سازی