

CFC PYTHON ENVIRONMENT: DEVELOPMENT OF WORKFLOW FOR INCLUDING EXTERNAL SOFTWARE INTO CFC PYTHON ENVIRONMENT

Objective

- GUI development for CFC centralized Python environment.
- Investigating the solutions for interaction with external programs and incorporation of the non-Python scripts with the focus on CFC in-house software.

The Estimated duration of the project is 10-12 weeks. This study does not address the databases and processing of output files.

Motivation

The proposed study includes two major parts: For an estimated period of 4 weeks, the approaches (see the Methodology chapter), to software developments and interaction with external programs/programming languages will be investigated.

The second part is dedicated to the generation of a complete and functional Python package and its user interface within an estimated period of 6 weeks.

The proposed study and the subsequent package development only serve as part of CFC software development. The workflow in this document is designed to resolve the uncertainties of each method. The knowledge and experience gained during this investigative period will shape the strategy toward efficient planning for future software developments as well as facilitating the management of the ongoing programming by individuals at CFC.

The Graphic User Interface is based on the concept of dashboarding. The selected dashboarding package (Streamlit) provides a strong interactive tool in Python that turns data into shareable web apps in minutes, which then may be accessed by any company employee through the company server. A visual and interactive interface will make the members more comfortable with Python scripting and encourage them to participate.

Methodology

Approach 1: The Input Generator:

In this approach, the Python Module generates the input text file. The user must submit the input file for calculation. The Python Module specifically acts as a helper agent to facilitate input generation.

Approach 2: Wrapper program:

The original script is “wrapped” in a Python environment. Although it may be possible to access the input command similar to approach 1, The entire process of input generation and execution is performed internally by Python (through a 3rd party package) and the original program, for instance, “CARSEC” in Fortran or the VB command within an Excel file will directly work as the calculation engine.

This approach requires further investigation.

Approach 3: Write in Python:

(Re)write the script directly in the native Python environment.

Approach No	Pros	Cons
1: Input generator	-Modifiable input in original environment.	-Remains dependent on Fortran/VB -Requires user interaction with the source code
2: Wrapper program	-Could utilize the code generated for approach No.1 -Optional interaction with the original environment (execute VB for Excel files, input submission in Linux).	- Remains dependent on Fortran/VB.
3: Rewrite the code in Python	-No dependency on the original program/language - Easier to maintain.	-requires debugging and testing.

Road Map

Stage 1) Evaluation of workable solutions

Estimated duration: 2 weeks.

The process of adoption of the Original Program by approaches 1 and 2 will be assessed using:

- One simple Fortran program, (preferably one of the simpler CFC programs).
- One Excel calculations sheet (with Visual Basic scripting).

The findings and experience gained at the end of this stage will be the key decision-making factor in the future implementation of non-Python scripts and external programs.

Stage 2) Development of a beta version of GUI

Estimated duration: 2 weeks.

A basic Graphical User Interface will be generated for the scripts of Stage 1 using the Package Streamlit. The selection of Streamlit is based on a previous broad assessment of available GUI development tools and specifically Dashboarding packages.

The Python scripts produced by the developer must intend to generate different styles for each GUI, this is to understand the possibilities of formatting and style and select the best fit for CFC applications.

At this stage, the developer team will present the results of Stages 1 and 2 as well as its recommendations regarding the selection of the methodology. The next stages will be followed accordingly.

Stage 3) Development of Python packages

Estimated duration: 4 weeks.

Depending on the approach selected in the previous stage, a Module/Package will be developed in this stage. The developer must ensure the possibility of future extension of the code as well as its compatibility with other – future – CFC Python Packages. Therefore, implementation of Python Classes (i.e., Object-Oriented Programming) is suggested.

Stage 4 Development of GUI

Estimated duration: 2 weeks.

In addition to extending the early version of the GUI, the possibilities of launching a private and centralised web-based interface will be investigated; Launching the applications on the CFC server is the favourable choice. However, another private server may be selected depending on accessibility and privacy. By the conclusion of this stage, the developed applications will be ready for independent reviews and minimal corrections if required.