Project report: Coupling ScimBa and Feel++

Helya Amiri, Rayen Tlili

Contents

1 Introduction

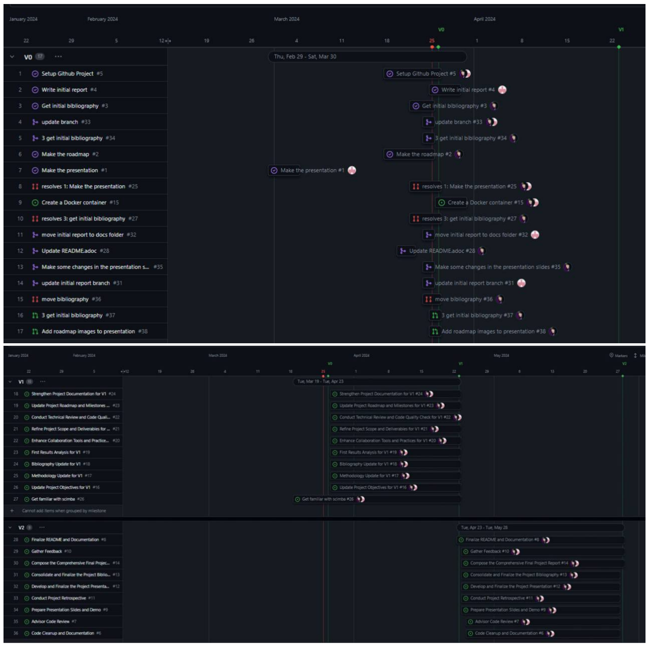
This report presents the objectives, approach, and roadmap for the coupling of ScimBa and Feel++ libraries. ScimBa is a project aimed at integrating machine learning techniques with traditional scientific computing methods, while Feel++ is a C++ implementation of Galerkin methods for solving partial differential equations (PDEs). The coupling of these two libraries is expected to enhance their capabilities and enable researchers to solve complex scientific problems more effectively.

1.1 Objectives

This project seeks to facilitate the coupling of ScimBa and Feel++. The primary objective is to establish loose coupling between ScimBa and Feel++, enabling efficient utilization of their respective strengths.

1.2 Roadmap

Github Roadmap



V0 17
Thu, Feb 29 - Sat, Mar 30

1 Setup Github Project #5
2 Write initial report #4
3 Get initial bibliography #3
4 update branch #33
5 3 get initial bibliography #34
6 Make the roadmap #2
7 Make the presentation #1
8 resolves 1: Make the presentation #25
9 Create a Docker container #15
10 resolves 3: get initial bibliography #27
11 move initial report to docs folder #32
12 Update README.adoc #28
13 Make some changes in the presentation s... #35
14 update initial report branch #31
15 move bibliography #36
16 3 get initial bibliography #37
17 Add roadmap images to presentation #38

V1 10
Tue, Mar 19 - Tue, Apr 23

18 Strengthen Project Documentation for V1 #24
19 Update Project Roadmap and Milestones ... #23
20 Conduct Technical Review and Code Quali... #22
21 Refine Project Scope and Deliverables for ... #21
22 Enhance Collaboration Tools and Practice... #20
23 First Results Analysis for V1 #19
24 Bibliography Update for V1 #18
25 Methodology Update for V1 #17
26 Update Project Objectives for V1 #16
27 Get familiar with scimba #26
Cannot add items when grouped by milestone

V2 9
Tue, Apr 23 - Tue, May 28

28 Finalize README and Documentation #8
29 Gather Feedback #10
30 Compose the Comprehensive Final Projec... #14
31 Consolidate and Finalize the Project Biblio... #13
32 Develop and Finalize the Project Presenta... #12
33 Conduct Project Retrospective #11
34 Prepare Presentation Slides and Demo #9
35 Advisor Code Review #7
36 Code Cleanup and Documentation #6

The general outline as of the start of the project includes the following goals:

1. **Analysis and Planning:** Analyze the functionalities of ScimBa and Feel++ to identify integration points. Develop a plan for implementing loose coupling between the two libraries.

2. **Implementation:** Implement loose coupling between ScimBa and Feel++ according to the planned approach. Develop methods for generating datasets using Feel++ and integrating them into ScimBa.

3. **Testing and Validation:** Test the integrated system to ensure that communication between ScimBa and Feel++ is seamless. Validate the effectiveness of the coupling by solving complex scientific problems.

4. **Documentation and Reporting:** Document the integration process, including interface definitions and communication protocols. Prepare a final report summarizing the project outcomes and lessons learned.

2 Getting started

2.1 Creating a Docker container

Creating a Docker container and image for the project offers these key advantages:

1. **Portability:** Run the project on any platform supporting Docker.

2. **Isolation:** Avoid conflicts with other software on the host system.

3. **Reproducibility:** Recreate the exact same environment whenever needed.

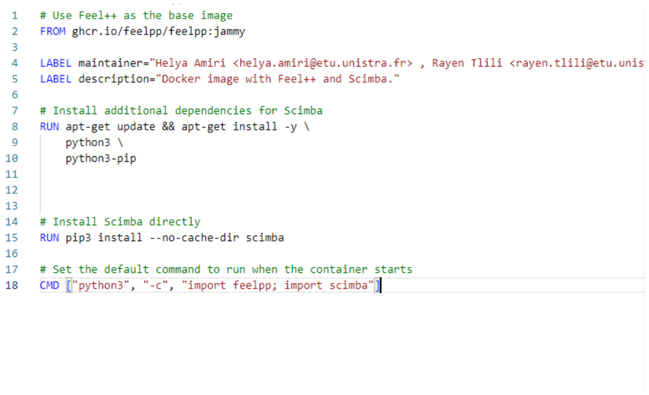
4. **Dependency Management:** Package all dependencies within the Docker image.

We're using Feel++ as the base for the Docker container adding the requirements and dependencies for Scimba.

This contains the latest version of Feel++ and Scimba and should be able to run these commands without error:

```
python
import feelpp
import scimba
```

Dockerfile

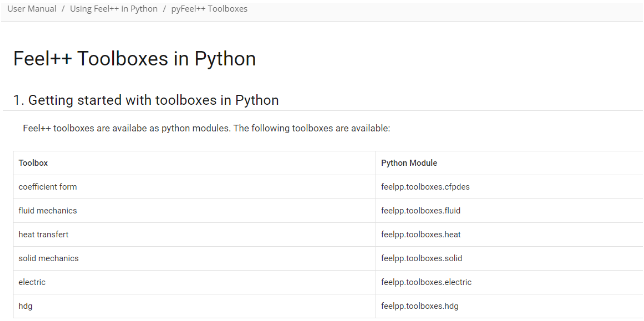


```
# Use Feel++ as the base image
FROM ghcr.io/feelpp/feelpp:jammy

LABEL maintainer="Helya Amiri <helya.amiri@etu.unistra.fr> , Rayen Tlili <rayen.tlili@etu.unis
LABEL description="Docker image with Feel++ and Scimba."

# Install additional dependencies for Scimba
RUN apt-get update && apt-get install -y \
    python3 \
    python3-pip



# Install Scimba directly
RUN pip3 install --no-cache-dir scimba

# Set the default command to run when the container starts
CMD ["python3", "-c", "import feelpp; import scimba"]
```

2.2 Exploring Feel++ toolboxes

As of the first meeting with the project supervisors, we've taken a look at the different toolboxes Feel++ has to offer in python:

.



User Manual / Using Feel++ in Python / pyFeel++ Toolboxes

Feel++ Toolboxes in Python

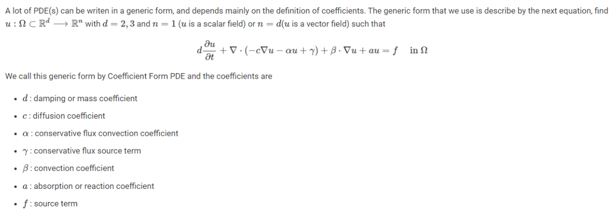
1. Getting started with toolboxes in Python

Feel++ toolboxes are availabe as python modules. The following toolboxes are available:

Toolbox	Python Module
coefficient form	feelpp.toolboxes.cfpdes
fluid mechanics	feelpp.toolboxes.fluid
heat transfert	feelpp.toolboxes.heat
solid mechanics	feelpp.toolboxes.solid
electric	feelpp.toolboxes.electric
hdg	feelpp.toolboxes.hdg

An interesting toolbox to start with is the **Coefficient Form PDEs**.

.



A lot of PDE(s) can be writen in a generic form, and depends mainly on the definition of coefficients. The generic form that we use is describe by the next equation, find $u : \Omega \subset \mathbb{R}^d \longrightarrow \mathbb{R}^n$ with $d = 2, 3$ and $n = 1$ (u is a scalar field) or $n = d$(u is a vector field) such that

$$d\frac{\partial u}{\partial t} + \nabla \cdot (-c\nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + au = f \quad \text{in } \Omega$$

We call this generic form by Coefficient Form PDE and the coefficients are

- d : damping or mass coefficient
- c : diffusion coefficient
- α : conservative flux convection coefficient
- γ : conservative flux source term
- β : convection coefficient
- a : absorption or reaction coefficient
- f : source term

.

This toolbox allows users to solve PDEs in a generic manner just by inputting the different coefficient values. Implementing an interface which, through Scimba, calls Feel++, solves the PDE and retrieves the solution would be very powerful as it enables us to approach solving for Laplacian and heat transfer problems for example.

3 Bibliography

- Feel++ Documentation: `https://docs.feelpp.org/user/latest/index.html`
- ScimBa Documentation: `https://sciml.gitlabpages.inria.fr/scimba/`
- Coupling (Computer Programming): `https://en.wikipedia.org/wiki/Coupling_(computer_programming)`
- Using feel++: `https://www.cemosis.fr/events/course-solving-pdes-with-feel/`