CHART-MagNum 2021 Yearly Report

Dr. Jasmin Smajic, Dr. Michał Maciejewski

Institute of Electromagnetic Fields, ETH Zürich

1. Introduction

The main goal of the project CHART Magnet Numerics (MagNum) project in 2021 was to propose and develop an initial version of a framework for design of Nb3Sn superconducting accelerator magnets. The framework implements Model-Based System Engineering (MBSE) concepts to streamline the modelling workflows and, at the same time, ensure model consistency, traceability, and reproducibility. In detail, we developed: (i) an Application Programming Interface (API) in python for handling model input and magnet geometry, as well as interacting with simulation software; (ii) Jupyter notebooks for encapsulating magnet models; (iii) Docker containers for isolating a simulation software and enabling cross-operational system execution; (iv) multi-model and multi-physics optimization platform; (v) GitLab templates for model versioning and continuous integration pipelines. In the following we describe the main achievements in each of the aforementioned project areas.

2. MagNum API

The MagNum API is organized into several modules. The geometry module provides data structures and algorithms for handling of 2D magnet geometries such as cos(Θ) and block-coil types following the definition from widely used ROXIE electromagnetic solver. In addition, the cos(Θ) geometry supports inputs with relative positioning angles and slots between coil layers. The module starts with a generic input and returns data structures with tool-dependent geometry definitions. The tool adapter module provides a programmatic interface to build models and control several solvers: (i) ROXIE for electromagnetic simulation; (ii) ANSYS APDL for mechanical modelling; (iii) custom adiabatic hot-spot temperature estimation.

With the MagNum API, a modelling activity starts with a standard input describing geometry and magnet parameters, continues with automatically created model inputs, and eventually ends with post-processed model outputs.

3. Jupyter Notebooks

The Magnum API is imported and called to perform modelling tasks in a python Jupyter notebook. A notebook contains source code, results of its execution such as graphs and printouts along with a documentation in a form of text, tables, equations, etc. A notebook starts with a parametric input for model geometry and parameters. Afterwards a model is built and executed with a numerical solver. Eventually, model results are post-processed, as well as relevant scalar figures of merits and artefact files are returned. As such a notebook is a view of a model for a particular study. A model view is saved as an interactive report book for versioning and reproducibility.

4. Docker Containers

We rely on Docker containers to provide a unified and isolated (for the sake of reproducibility) modelling environment across popular operating systems. Unlike a virtual machine, a Docker container does not require installation of an operating system on a host OS. Instead, the Docker engine is handling information exchange between a container and a host OS. As a result, containers are lightweight and support all modern operating systems. A Docker image encapsulates a simulation tool with dependencies and exposes an API. In the scope of the project, we created a dedicated Docker image for ROXIE electromagnetic solver.

5. Initial MBSE Framework Architecture

The three aforementioned pillars form a foundation for the initial MBSE framework architecture; see Figure 1. The basic building block is a notebook-based model (also referred to as an MBSE model). An MBSE model executes API functions to interact with Docker-isolated solvers in order to create a particular study view. Furthermore, in a similar manner measurement data is queried and integrated into an analysis. The proposed architecture allows for an analysis of heterogeneous data sources.

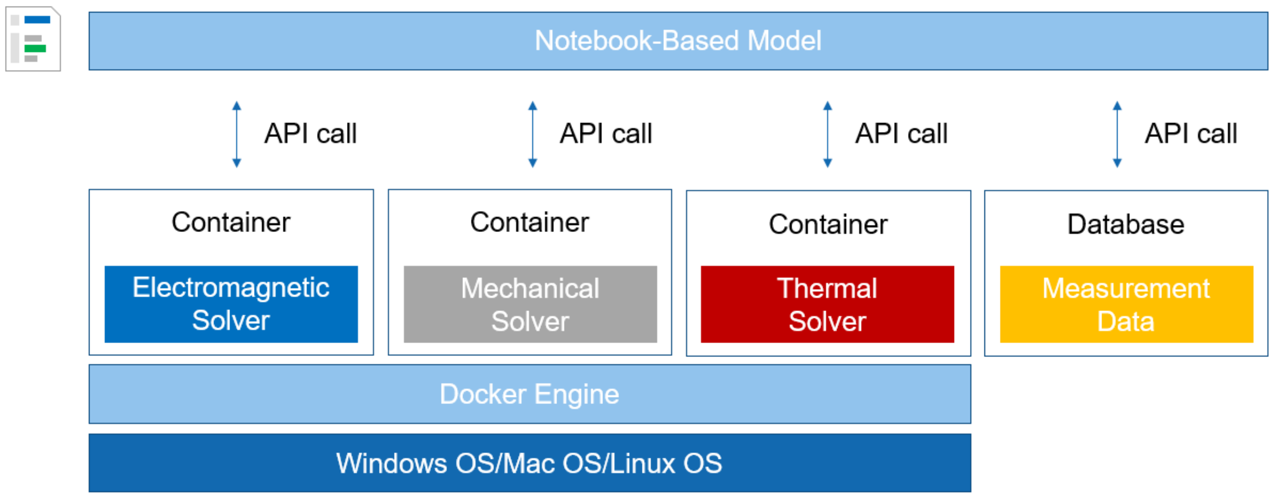

Figure 1. Notebook-model interaction with solvers and measurement databases

MBSE methodology shifts the focus from documents to models in system design: (i) models provide an abstraction of a system; (ii) models generate views with relevant data; (iii) models are queried for data by other models; (iv) models are versioned for life-cycle management, consistency and traceability.

6. MBSE Trade Study

The relevant notebook models are combined together to perform a trade study in order to obtain a system value. We created a multi-objective optimization method to find an optimal geometry of a high field superconducting magnet as shown in Figure 2. For a given operating field and magnet aperture, we aim to minimize: (i) absolute value of the magnetic field multipoles (field quality); (ii) position on the superconductor load line (safety margin); (iii) peak stress in the coil (superconductor degradation); (iv) adiabatic hot-spot temperature. The objective is obtained by adjusting positioning angle, inclination angle and a number of conductors per block. For a wide range of the design variables, may lead to an inconsistent geometry.

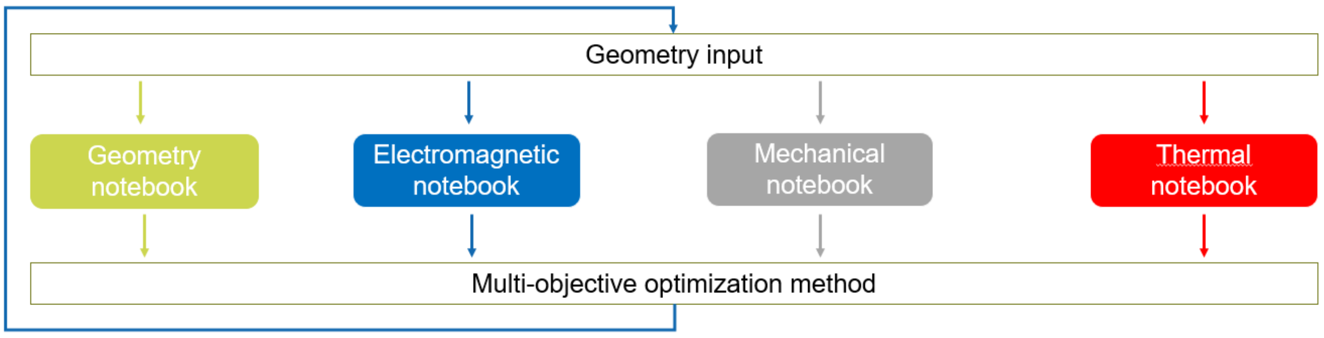

Figure 2. Multi-objective optimization method

7. GitLab Workflows

To ensure system design versioning and promote consistent collaboration, the inputs and notebook models we rely on GitLab platform. Each design change constitutes a commit. The main branch keeps track of the reference design version. Relevant design versions are labelled with tags. A design branch is suited for indicating design variants, e.g., an as-built version, a Conceptual Design Report version. Furthermore, the modelling notebooks are concatenated into multi-model GitLab pipelines. The pipelines are executed with each design change and on regular intervals in order to guarantee reproducibility. A visualization of this idea is depicted in Figure 3.

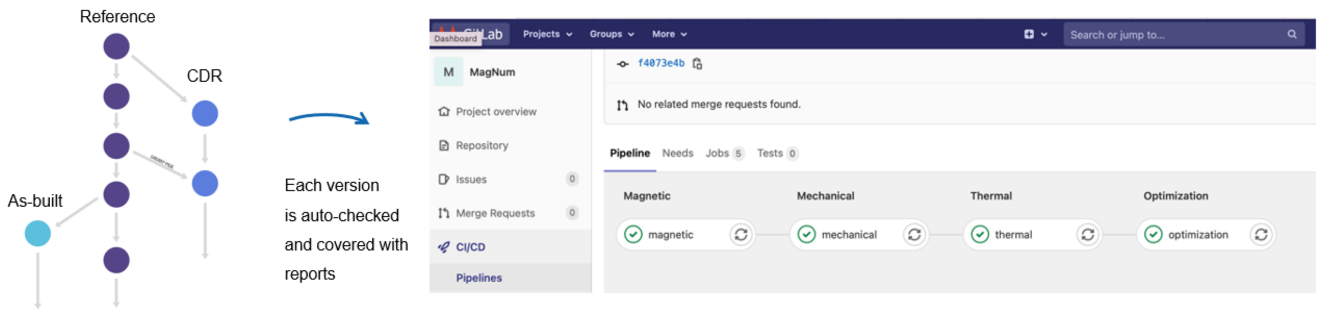

Figure 3. A system design versioning tree with modelling pipeline execution

8. Conclusion and Outlook

In the first year we achieved all expected outcomes. In particular, we proposed an initial MBSE platform architecture composed of an API, notebooks, and docker images. The notebooks were combined into a trade study for optimization of superconducting magnet geometry. Furthermore, the models are combined into pipelines and versioned with a standard GitLab platform.

The first version of the framework was already used by magnet engineers at Paul Scherrer Institute and Lawrence Berkeley National Laboratory. The obtained results were presented during a technical review at CERN [1, 2]. The framework architecture was presented at a COMPUMAG 2021 conference [3]. The details of the trade study were presented at Magnet Technology 2021 conference [4]. Note that, although the multi-model-optimization was handled with our MBSE framework, the framework itself is applicable far beyond a magnet

The MBSE framework implements already several relevant concepts, i.e., model view, model versioning, reporting, programmatic solver interface, etc. The next step is to enable model queries. Following this approach, models can obtain both figures of merit and artefacts from one another; see Figure 4. In a naïve approach this may lead to an excessive execution of models. Therefore, in the second year we plan to introduce a model registry to keep track of model dependencies along with a cache database for retrieving already solved models.

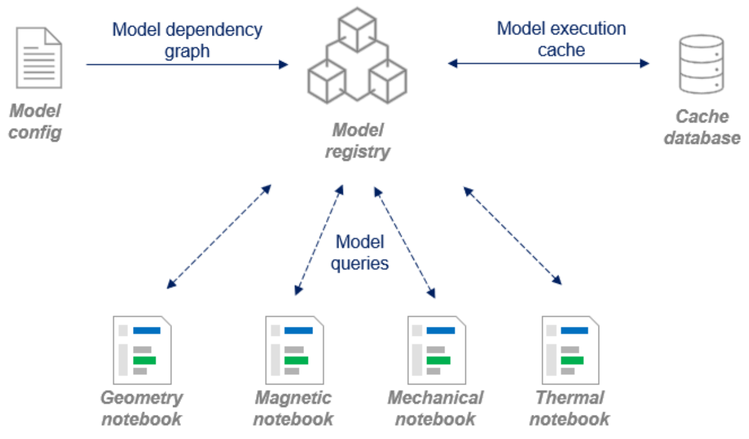

Figure 4. An MBSE framework architecture

References

[1] CHART2/MagDev Results and Roadmap - Technical Review, 13-14.12.2021, CERN, Geneva, Switzerland, https://indico.psi.ch/event/11955/

[2] Magnet design report created for Technical Review https://cern.ch/auchmann/chart/magnum

[3] M. Maciejewski, J. Smajic, L. Leuthold, B. Auchmann, D. Martins, G. Vallone, Model-Based Workflows for Multi-Physics Design Optimization of Superconducting Accelerator Magnets, IEEE Transactions on Magnetics, 2022, in preparation

[4] G. Vallone, B. Auchmann, M. Maciejewski, Magneto-Mechanical Optimization of Cross-Sections for cos(Θ) Accelerator Magnets, IEEE Transactions on Applied Superconductivity, 2022, in preparation