



HAL
open science

MagnetDB a platform for the simulation of High Field Magnet

Christophe Trophime, Jeremie Muzet, Debray Francois, Christophe Prud'Homme, Vincent Chabannes

► **To cite this version:**

Christophe Trophime, Jeremie Muzet, Debray Francois, Christophe Prud'Homme, Vincent Chabannes. MagnetDB a platform for the simulation of High Field Magnet. MT28 International Conference on Magnet Technologies, ITER Organization, Sep 2023, Aix - en - Provence, France. hal-04220904

HAL Id: hal-04220904

<https://cnrs.hal.science/hal-04220904v1>

Submitted on 28 Sep 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MagnetDB a platform for the simulation of High Field Magnet

Christophe Trophime, Jérémie Muzet, François Debray, Christophe PrudHomme and Vincent Chabannes

Abstract—We present a platform, MagnetDB designed to simplify and automate the simulation of high field magnets. The simulation available range from semi-analytical model for the magnetic field in the user experimental zone to multi-physic finite element model. The system architecture is composed of a web front-end with a python back-end that interact with a database containing a formal description of our magnet derived from our operating mode. The system focus on resistive magnets. We give an overview of the system with a focus on the templating engine to generate files for simulations setup. More advance use of the system with a python command line interface module is explored.

Index Terms—high field magnet, electromagnet, finite element, software architecture.

I. INTRODUCTION

THE French magnet user facility of the European Magnetic Field Laboratory (EMFL), namely the Laboratoire National des Champs Magnétiques Intenses (LNCMI), enable researchers to perform experiments in the highest magnetic field. DC magnetic fields up to 37 T are provided at the Grenoble site and pulsed fields up to 100 T at Toulouse. In this paper, we focus on DC resistive magnets from Grenoble site.

In collaboration with Cemosis of Uni. Strasbourg., we develop a software tool chain, *HiFiMagnet* [1], to design and model our magnets. *HiFiMagnet* is composed of in-house developments for the initial design and of finite element models based on *Feel++* library [2] for numerical multi-physics simulations that range from semi-analytical 1D to 3D fully coupled models. They allow covering multiple usage from the Magnet designer to the User. *HiFiMagnet* can generate CAD geometry and mesh for the studied magnets using *Salome* [3]. Depending on the model size, simulations can be run either on standard laptop or HPC facilities thanks to *Feel++* transparent management of parallelism. The setup of simulations is achieved with human-readable configuration files, coined as the input dataset in the sequel. Still, this process remains difficult for users not familiar nor with *Feel++*, nor with running jobs on HPC.

To address this point, we develop a platform to simplify and automate the setup of simulations as well as running the simulations. The platform consists mainly of a web frontend associated with an API to recover magnet data from an

SQL database and to feed them into a templating system for generating the input dataset associated to a selected model. Finally, a worker is in charge of running the simulation on a selected computer and store the results in a S3 storage system.

Beside this goal, the platform also offers the possibility to aggregate all the code developed for modeling the resistive magnets – ranging from semi-analytical axisymmetric models for predicting the magnetic field to full scale 3D multi-physic finite element model. On top of that, it allows offering new services for both magnet designers and users. So far, the platform access is still limited to the LNCMI staff.

Similar platforms are actively developed, mostly in the HTS superconductivity community, for educational [4] and magnet development [5] [6]. With *MagnetDB* platform, we aim to provide similar simulation services for our high field magnets, limited to the study of solenoid configurations. We also design the platform to exploit data from the magnet control system (MSC) to mimic magnet in real operating conditions.

The paper will be organized as follows. In a first section, we briefly present the architecture of *MagnetDB*. The central part of the system is the database representing our magnets that is detailed in section III. Then, we introduce the template engine to perform the simulation setup and the worker that actually launch the simulations. Finally, we illustrate the use of *MagnetDB* for modeling Bitter magnets from a command line interface that allows for more advanced workflow.

II. PLATFORM ARCHITECTURE

MagnetDB platform consists mostly in a web front-end and a python back-end app (see fig. 1). The access to the front-end is controlled by a Single Sign On (SSO) system [7] that allows for federate authentication. So far, the front-end is accessible only for LNCMI staff – namely to the laboratory *ldap* directory service. The SSO system generates a token for each user that can be also used for accessing the platform.

The python back-end is an API that (i) interacts with the SQL database that holds all information about the actual magnets, (ii) fetch magnet data from the database, send them to the web front-end to be displayed (e.g. magnet definitions – both in terms of components or material properties; magnet field maps; operating conditions for each run attached to a given magnet), (iii) performs simulation setup on demand, (iv) prepare the commands to execute for running the simulation remotely and (v) stores/loads data on a S3 storage service (e.g. CAD and mesh files attached to magnet, simulations results).

A worker service has been added to dispatch simulation setup and jobs in a queue to enable asynchronous treatment

C. Trophime, J. Muzet and F. Debray, Laboratoire National des Champs Magnétiques Intenses CNRS, Univ. de Grenoble, UPS, Insa, Grenoble, France
C. PrudHomme and V. Chabannes, Cemosis, IRMA, Univ. de Strasbourg, CNRS, Strasbourg, France

Manuscript received September 19, 2023.

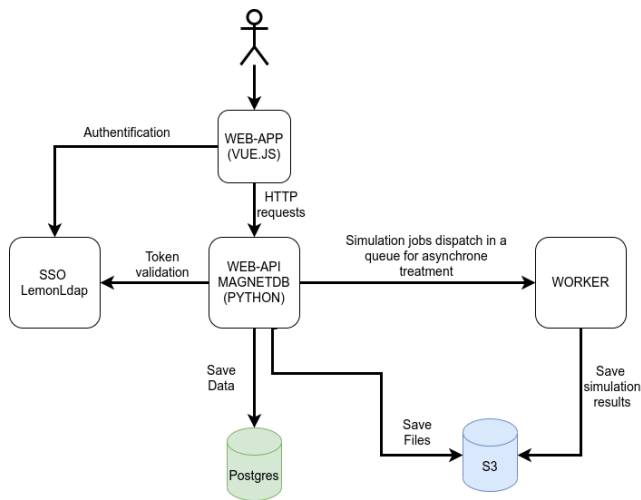


Fig. 1. MagnetDB architecture.

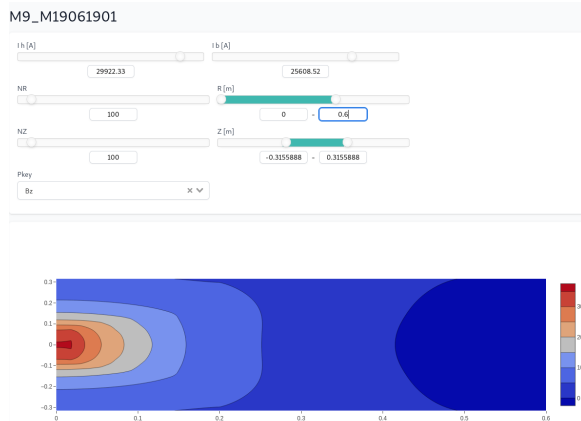


Fig. 2. *MagnetDB*: Magnet Field profile in (Or, Oz) plane for a given magnet. The user can define the currents in each insert composing the magnet, z and r range. The magnetic field is computed from a semi-analytical axi model.

since these operations may require important time resources, especially for 3D model. Simulation jobs can be run remotely on any computer resources defined by the user. All that is needed at this point is some credentials to log on to the remote computer, along with the computer's main characteristics (number of CPUs, ..). *MagnetDB* creates an ssh key for each computer defined by the user. To seamlessly run the simulation on the remote machine, the user simply needs to copy the ssh key to the remote machine. The *Feel++* app to be run on the target machine shall be shipped as an HPC container [8], installed prior to any operations on the target machine. In a future *MagnetDB* version, we plan to use a registry service [9] to automatically retrieve the container if not present on the remote computer.

On top of *MagnetDB*, we also implement a python module to interact with the platform in a command line interface (CLI) mode. The token issued by SSO – specific for each user – is needed to operate this CLI as will be shown in section V. This mode allows exploring new uses of the platform and creating new simulation workflows.

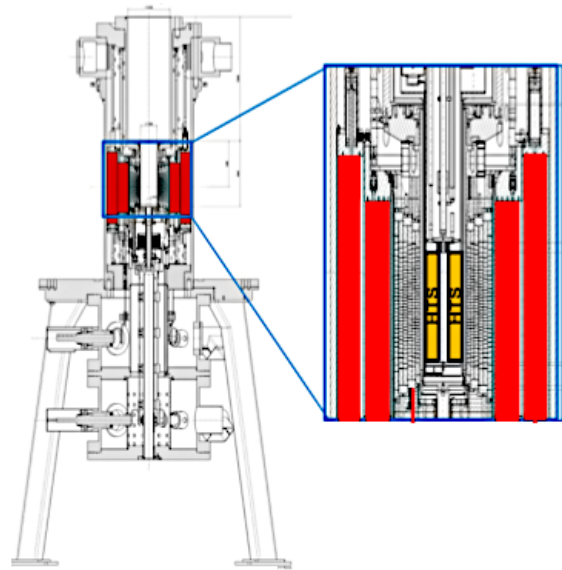


Fig. 3. Cut View of an experimental site: in *MagnetDB* terminology, the **site** is composed of an HTS **magnet** in “orange”, a poly-helices **magnet** in “gray” and Bitter **magnet** with two **parts** in “red”. Poly-Helices magnet is a specific technology of LNCMI, see for [11] details.

The *MagnetDB* platform has been configured to start using docker-compose that automatically launch each service (SSO system, Database, front-end, back-end). The services accessible from internet are routed using *traefik*[10] with ssl for security issues. A service to monitor and manage the database is also provided for initial setup of the SQL engine.

III. DATABASE STRUCTURE

The database (*Postgres* service in fig. II) is the actual central component of *MagnetDB*. Its structure reflects our operating mode. Magnets are fitted into a housing that enable to cool them down by a high speed water flow. The cooling system is actually similar to the one for nuclear plant - i.e. a primary cooling loop to cool down the magnet powered by a 24 MW installation and a secondary loop to cool down the water going out of the magnet. The resistive high field magnet consist of 2 inserts powered in parallel. The inner insert is made of poly-helices magnet and the outer of Bitter magnets [11]. Superconductor magnet can be “added” to this setup to reach higher fields: Hybrid project [12] with an LTC outsert which is designed to reached 43 tesla (this magnet is currently tested), an HTS insert [13] that has reached 32.5 tesla - see fig 3.

In *MagnetDB* terminology, we define a **site** as a set of **magnet**. In turn, a **magnet** is composed of **parts** (see fig. 4). A **part** is a geometry with a **material**. The **part** geometry is defined for the moment as yaml file, stored in the S3 service. In a near future version, the geometry will also be defined as a json database object. The physical properties are introduced as fields of a **material** object. Again, we have limited ourselves to scalar properties. This will be extended to support tensor as well as tabulated properties. For superconductor material, we envision to use a json db object that contains parameters and fit function for $\mathbf{J}_c(\mathbf{B}, T)$ [14].

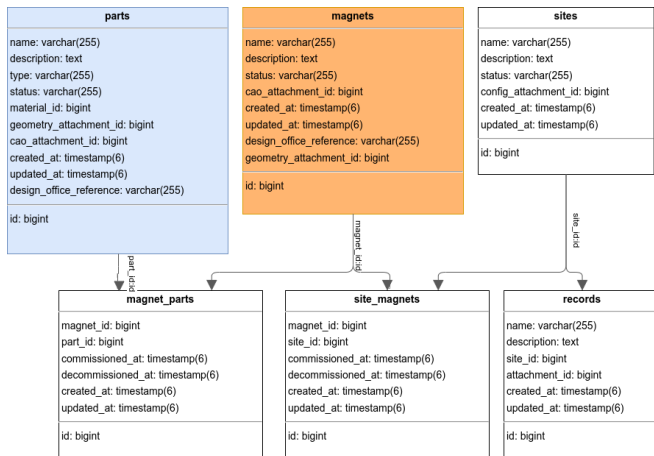


Fig. 4. MagnetDB database structure.

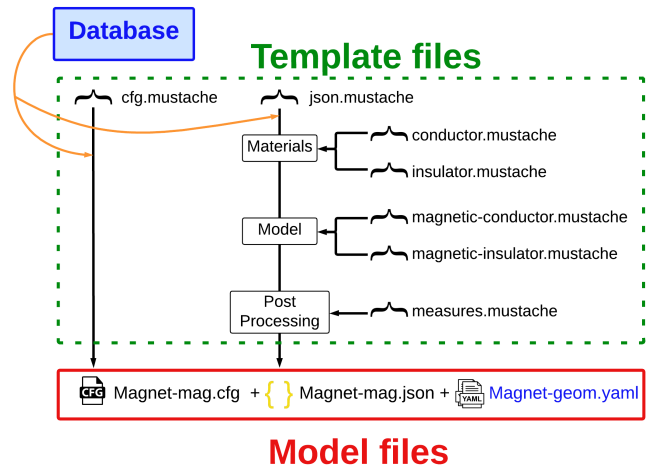
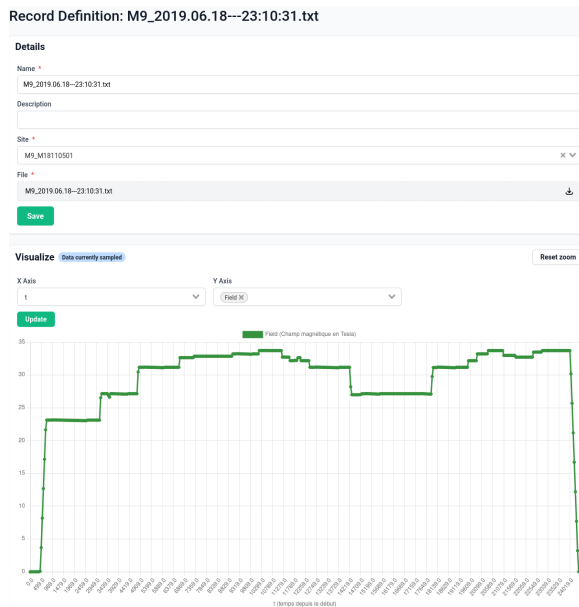


Fig. 6. Schematic view of the template setup engine.

Fig. 5. MagnetDB: view of a record. Here, the magnetic field $B_z(0,0)$ is displayed versus time for a given record. The user can select the fields to view by selecting it in “X-Axis” and “Y-Axis” sliding menu. The data correspond to a “magnet run”.

Associated db tables are used to respectively keep track of the **parts** of a **magnet** and of the **magnets** composing a **site**. This provides a way to have a history of either use **part** or **magnet**. Indeed, when a **magnet** breaks, only certain **parts** are replaced. In our terminology, a **magnet** failure means that the corresponding **site** is retired, which corresponds to a change of status in the db. Operational data from the MSC - namely **record** in MagnetDB - are attached to each operational **site**.

Beside the magnet formal description, the database also contains specific tables for monitoring database operations, for controlling what a user can do. We have classically implemented “Admin”, “User” and “Guest” roles. “Guest” can only view data, “User” can add/modify data.

IV. TEMPLATING SETUP SYSTEM

As briefly stated in section II, the simulation setup is performed by call to the python API. More precisely, we developed a python module – namely *magnetsetup* - that generate files needed for running a simulation with *feel++* apps using a *mustache* template engine (fig. 6).

A form is provided to select either the **magnet**, either the **site** to model, the actual simulation to perform and several other fields depending on the selected simulation type (see fig. 7). When the user click on the “Save” button, the API retrieve data from db for the selected object and generate the actual simulation files needed by sending appropriate data to the template engine. Once the files are generated, a **simulation** object is created and the user can proceed with setup by clicking on the corresponding button. Then, running the simulation can be started just by clicking on the “Run” button. At this stage, the user is asked to select a computer before actually running the simulation. In case of failure at any steps, the user is notified – see the “setup pending” in fig. 8 which stands for setup operation in progress – and may retrieve log files of the failed operation for debugging.

Template files per numerical model are defined as a dictionary in a magnetsetup json file, loaded when the system is powered on.

V. ADVANCE USE WITH CLI MODE

We also provide a python module to interact with MagnetDB in CLI and to perform more advanced simulation workflow. To enable this mode, the user just need to get the API key generated when first logging into MagnetDB and define an environment variable that points to this token.

To illustrate this mode, we will mimic the commissioning procedure of a Bitter magnet. In this procedure, the cooling water flow rate Q is controlled by the current I in the Bitter magnet. As a consequence, we need to estimate for each current the heat exchange coefficient h and the water temperature rise ΔT_w to properly model the magnet cooling. h is computed using heuristic correlation [15] and ΔT_w is

Fig. 7. *MagnetDB*: form to setup a **simulation**. Here, we create a static linear thermo-electric model for an axisymmetrical geometry with a “mean” cooling model. The cooling model does not consider each cooling channels separately.

Fig. 8. *MagnetDB*: launch setup and run a **simulation**. The model is the same as the one defined in fig. 7.

obtained from a power balance. Consequently, we perform an iterative thermo-electro-magnetic simulation as h and ΔT_w are function of I . From a practical point of view, this model - *python_magnetnetworkflows.cli* - has been implemented using the cli module and *Feel++* python interface.

Listing 1. Commissioning workflow in CLI

```
python -m python_magnetapi.cli compute \
  --mtype magnet \
  --name M10Bitters \
  --flow_params \
python -m python_magnetapi.cli setup \
  --mtype magnet \
  --name M10Bitters \
  --method cfpdes --static --geometry Axi \
  --model thmag_hcurl --cooling mean \
  --nonlinear
singularity exec magnetgmsf \
  python3 -m python_magnetgmsf.cli \
  --wd data/geometries M10Bitters.yaml \
  --mesh
python -m python_magnetnetworkflows.cli \
  --mdata '{"M10Bitters": { \
    "value":17197, \
```

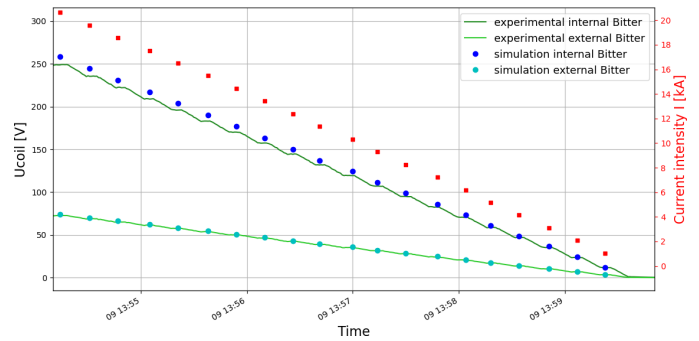


Fig. 9. *MagnetDB*: Comparisons of computed and measured Voltage taps during magnet commissioning. Each plateau corresponds to a given current (see legend on the right side)

```
"flow": "M10Bitters-flow_params.json"
}}' \
M10Bitters-sim.cfg \
--cooling mean
```

In a near future, we plan to run the simulation workflow each time a new magnet is defined in the system. The characteristics $R(I)$, $L(I)$, $h(I)$, $\Delta T_w(I)$ as well as the waterflow params could be attached to each **magnet** object. They could serve later as inputs for a simple circuit model to mimic in real-time the transient in operating conditions.

VI. CONCLUSIONS

We have developed *MagnetDB*, a platform for the simulation of our resistive magnets. It consists of a database that provides a formal description of the magnets and an API interface that allows automatic simulation setup. A worker service is used to seamlessly launch the simulation on a remote computer. Users can actually run simulation ranging from semi-analytical 1D model to multi-physics Axi or 3D finite element model from a web front end. Right now, the access to the platform is limited to LNCMI staff, but it may be opened for our users in a near future. In a near future, we plan to add a post-processing service for magnet designer to explore in detail simulation results, some basic stats service for e.g magnet consumption.

Data issued from the MSC are accessible from the front-end. This allows to model our magnets in real operating conditions through the use of a python CLI for interacting with *MagnetDB*. We foresee to extend this mode to adopt a systemic approach for more realistic simulations. This would be the first toward magnet digital twin to get more insights of magnet part life-time from the real “stress” history during its operation; coupling the simulations with the actual MSC system for early failure detection; estimating the water flow temperature rise from the user experiment in the prospect of fatal heat recovery project [16].

ACKNOWLEDGMENT

The authors would like to thank Remi Caumette for his help in implementing *MagnetDB* front-end and back-end.

REFERENCES

- [1] C. Daversin, C. Prudhomme, and C. Trophime, "Full three-dimensional multiphysics model of high-field polyhelices magnets," *IEEE Transactions on Applied Superconductivity*, vol. 26, no. 4, pp. 1–4, 2016.
- [2] Prud'homme, Christophe, Chabannes, Vincent, Doyeux, Vincent, Ismail, Mourad, Samake, Abdoulaye, and Pena, Goncalo, "Feel++ : A computational framework for galerkin methods and advanced numerical methods," *ESAIM: Proc.*, vol. 38, pp. 429–455, 2012. [Online]. Available: <https://doi.org/10.1051/proc/201238024>
- [3] "Opensalome: The open source platform for numerical simulation," Available at <https://www.salome-platform.org/>.
- [4] N. Riva, F. Grilli, and B. Dutoit, "AURORA: Learning Superconductivity Through Apps," in *7th International Workshop on Numerical Modelling of High Temperature Superconductors (HTS 2020)*. Nancy (Virtual), France: Kévin Berger (Université de Lorraine - GREEN), Jun. 2021. [Online]. Available: <https://hal.science/hal-03295937>
- [5] L. Bortot, B. Auchmann, I. C. Garcia, A. M. F. Navarro, M. Maciejewski, M. Mentink, M. Prioli, E. Ravaioli, S. Schps, and A. P. Verweij, "Steam: A hierarchical cosimulation framework for superconducting accelerator magnet circuits," *IEEE Transactions on Applied Superconductivity*, vol. 28, no. 3, pp. 1–6, 2018.
- [6] "Quanscient allsolve: a multiphysics simulation software," Available at <https://quanscient.com/>.
- [7] C. Maudoux and S. Boumerdassi, "LemonLDAP::NG a full AAA free open source WebSSO solution," in *2022 IEEE 11th International Conference on Cloud Networking (CloudNet)*. IEEE, nov 2022. [Online]. Available: <https://doi.org/10.1109%2Fcloudnet55617.2022.9978777>
- [8] G. M. Kurtzer, V. Sochat, and M. W. Bauer, "Singularity: Scientific containers for mobility of compute," *PLOS ONE*, vol. 12, no. 5, pp. 1–20, 05 2017. [Online]. Available: <https://doi.org/10.1371/journal.pone.0177459>
- [9] V. Sochat, "Singularity registry: Open source registry for singularity images," *Journal of Open Source Software*, vol. 2, no. 18, p. 426, 2017. [Online]. Available: <https://doi.org/10.21105/joss.00426>
- [10] "Traefik: an open-source edge router," Available at <https://doc.traefik.io/traefik>.
- [11] F. Debray, J. Dumas, C. Trophime, and N. Vidal, "Dc high field magnets at the Incmi," *IEEE Transactions on Applied Superconductivity*, vol. 22, no. 3, pp. 4 301 804–4 301 804, June 2012.
- [12] P. Pugnati *et al.*, "Commissioning of the 43+t grenoble hybrid magnet," *IEEE Transactions on Applied Superconductivity*, p. to appear in, 2023.
- [13] P. Fazilleau, X. Chaud, F. Debray, T. Lécresse, and J. Song, "38 mm diameter cold bore metal-as-insulation hts insert reached 32.5 t in a background magnetic field generated by resistive magnet," *Cryogenics*, vol. 106, p. 103053, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0011227519304291>
- [14] J. Muzet, c. Trophime, X. Chaud, C. PrudHomme, and V. . Chabannes, "2d axisymmetric modeling of the hts insert nougat in a background magnetic field generated by resistive magnet," *IEEE Transactions on Applied Superconductivity*, p. to appear in, 2023.
- [15] D. B. Montgomery, *Solenoid Magnet Design*. New-York.: Wiley-Interscience, 1969.
- [16] "Projet valocal: Valorisation des calories issues du procédé de production des champs magnétiques intenses," Available at https://www.celluleenergie.cnrs.fr/wp-content/uploads/2017/11/projet_valocal.pdf.