Quench Velocity-Based Approach in ANSYS APDL

Michał Wilczek
mailto:michal.wilczek@cern.ch

Abstract

This report presents the workflow of the quench velocity-based approach implemented in Python and ANSYS APDL. The method was developed to simulate the longitudinal and the turn-to-turn quench propagation in high-order corrector magnets. As illustrated in Fig. 1, there are four steps necessary for conducting the multi-strand analysis of a high-order corrector with the quench velocity-based approach. It is required to conduct a set of magnetic analyses, and a set of 1D quench simulations on simplified numerical models to predict both the quench velocity and the minimum propagation zone in the superconducting coil. When all three initial steps are completed, the quench simulation of the high-order corrector can be performed including the magnet discharge.

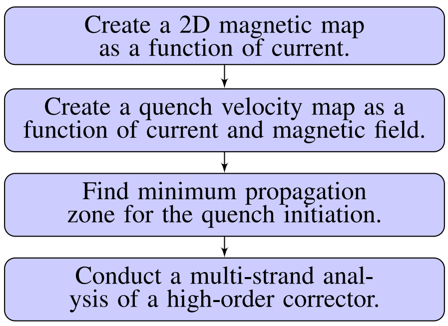

Figure 1: Analysis workflow for the quench analysis of a high-order corrector.

Moreover, the report describes how the ANSYS APDL simulations should be launched on the CERN cluster. The possible software development for the quench velocity-based approach is discussed as well.

Geneva, Switzerland
February 24, 2020

Contents

Chapter 1

Installation Prerequisites

In this chapter, the installation prerequisites are discussed to:

1. run the ANSYS simulations with the quench velocity-based approach;
2. rerun the set of exemplary ANSYS simulations illustration the developed tool.

1.1 General software requirements

In order to perform the ANSYS simulations with the quench velocity-based approach on a personal computer, the user should follow the given steps:

1. install ANSYS software (version 2019 R1);
2. install Python (version 3.6);
3. download the Python package for the quench velocity-based approach given under the following link `https://gitlab.cern.ch/steam/steam-ansys-modelling`;
4. import external libraries used in the Python package for the quench velocity-based approach.

There are two tools used from the ANSYS package: (*i*) ANSYS APDL, (*ii*) Design Modeler from ANSYS Workbench. ANSYS APDL is necessary to run the simulation with the quench velocity-based approach. Design Modeler is an additional tool required for the magnetic part described in Chapter 2. The Python code was developed in PyCharm (version 2018.3.4), being the commercial integrated development environment (IDE). The user is not obliged to use this tool. Nevertheless, another tool able to interpret the Python language is required in such a case.

1.2 Requirements for the exemplary ANSYS of simulations

The exemplary simulations are stored in two following repositories:

1. `\\cern.ch\dfs\Workspaces\a\ANSYS_Modelling\1_quench_velocity_modelling\documentation\Test_Simulation_Workflow`;
2. `https://cernbox.cern.ch/index.php/s/ViJX5rH5eLnKP0t/download`.

The first path presents an internal directory for the project to which the access is restricted. The second path is public. In case of any problem with accessing this link, please, contact Michał Maciejewski (`mailto:michal.maciejewski@cern.ch`).

It is assumed in this manual that the files are used as if they were stored in the first repository. Therefore, all the paths, which are specified in the following chapters, relate to the first directory. This is usually not the case for the individual use of the programme. In such a case it is required to specify the directory where the files are stored on a personal computer.

Chapter 2

Magnetic 2D Analysis

The 2D magnetic field analysis serves for obtaining the map of the magnetic field strength in the coil for different values of operating current. Two tools are used within the studies: ANSYS Workbench and ANSYS APDL. The former serves for creating the 2D geometry, and the latter is used to perform the magnetic field simulations. For more information about conducting magnetic field analyses in ANSYS APDL for this particular case, please refer to [1, p. 77-81].

The files required to rerun the simulations are stored in `\\cern.ch\dfs\Workspaces\a\ANSYS_Modelling\1_quench_velocity_modelling\documentation\Test_Simulation_Workflow\1_analysis_magnetic`

2.1 Creating the geometry in ANSYS Workbench

ANSYS Workbench is used because the iron yoke, being a part of the skew quadrupole geometry, was received as an external step file. It is not possible to open a step file in ANSYS APDL. Therefore, the geometry was created in Design Modeller of ANSYS Workbench and exported to ANSYS APDL. The geometry creation steps in ANSYS Workbench are as follows:

1. Import the step file with the iron yoke geometry to Design Modeller. The step file with the iron yoke can be found in `\input\skew_quad_iron_yoke.stp`.
2. Create the remaining part of the 2D high-order corrector geometry in Design Modeller. Figure 2.1 presents the final one-eighth of the skew quadrupole geometry. The geometry is reduced due to available three planes of symmetry in the geometry. It is recommended to mesh the geometry in ANSYS APDL after the exporting procedure. For more explanations why the meshing step is performed in ANSYS APDL rather than in ANSYS Workbench, please, read the private communication with ANSYS support presented in `\additional_help`.

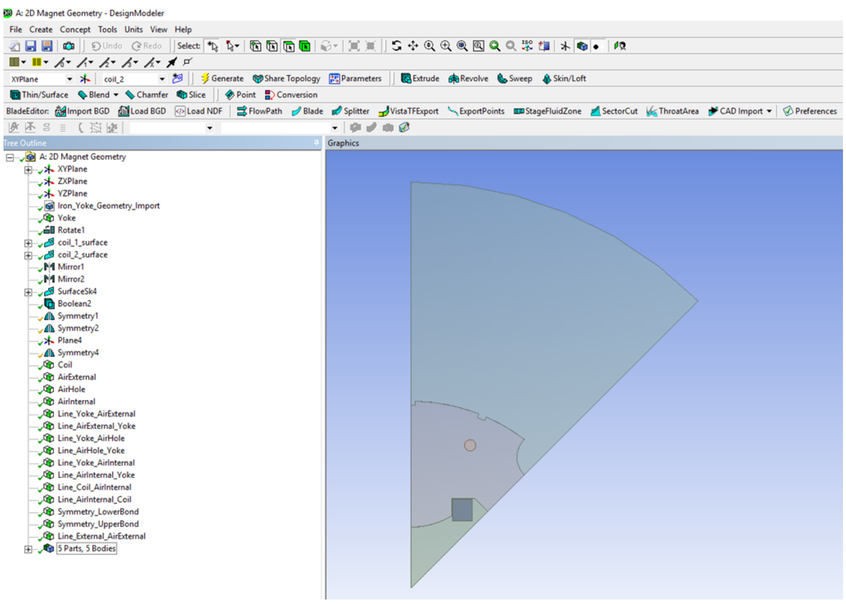

Figure 2.1: Final skew quadrupole geometry in Design Modeller of ANSYS Workbench.

3. As soon as the geometry is prepared in Design Modeller, the user should add the additional module in GUI of ANSYS Workbench, called Mechanical APDL as illustrated in Fig. 2.2. By clicking, “Edit in Mechanical APDL”, the geometry is easily exported to ANSYS APDL, and the ANSYS APDL window should open at this moment.

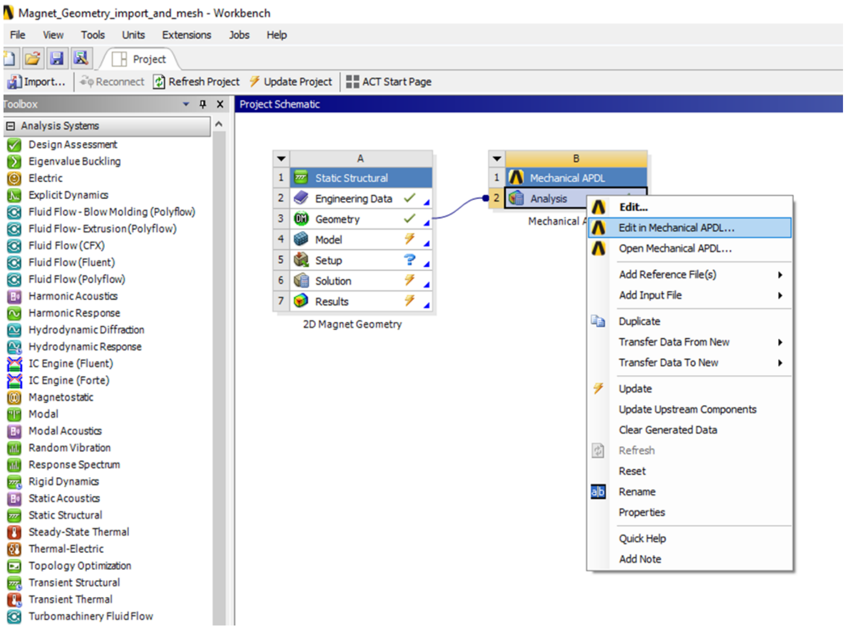

Figure 2.2: Exporting procedure of Workbench geometry to ANSYS APDL.

4. In order to verify whether the geometry is properly exported, the user should type “LPLOT” command in the command window of ANSYS APDL as shown in Fig. 2.3. If the geometry contour lines appear after typing the command, it means that the Design Modeler geometry is properly exported to ANSYS APDL.

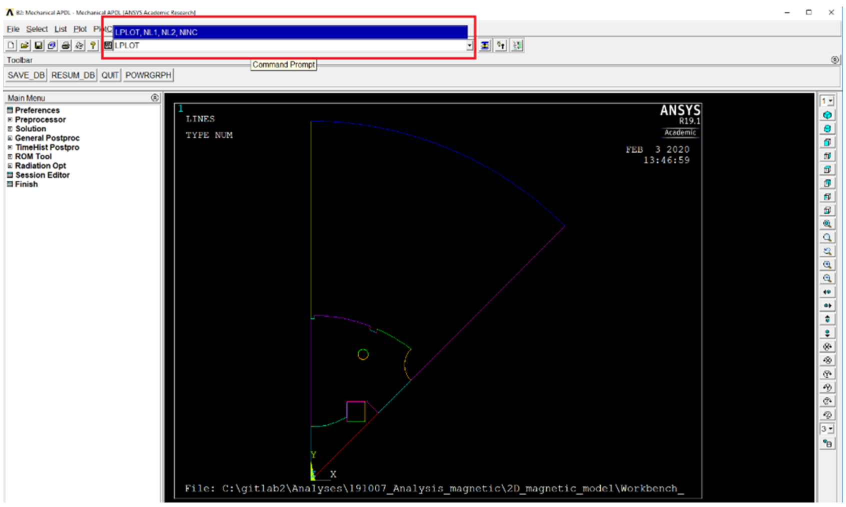

Figure 2.3: Illustration of APDL geometry and command to invoke its contour.

5. In the last step, the geometry should be saved in ANSYS APDL as a .db file, being a typical extension for the ANSYS APDL project.

2.2 Magnetic Analysis in ANSYS APDL

Depending on the step, from which the user is willing to rerun the magnetic analysis, there are four input files given in the folder `\input`:

1. `skew_quad_iron_yoke.stp`, being the imported step file to Design Modeller;
2. `Workbench_geometry_skew_quad.wbpz`, representing the entire skew quadrupole geometry prepared in ANSYS Workbench;

3. `Skew_Quad.db`, corresponding to the skew quadrupole geometry saved in ANSYS APDL;
4. `2D_MAGNETIC_ANALYSIS.inp`, being the ANSYS APDL script prepared to execute the magnetic analysis.

In order to conduct the magnetic analysis in ANSYS APDL, two last files from the list above are only required. Both are copied to the folder `\output`. The user can modify the parameters in the file `2D_MAGNETIC_ANALYSIS.inp` as presented in Fig. 2.4. It is recommended to run the simulations with current_init, current_max, and current_step being integer values.

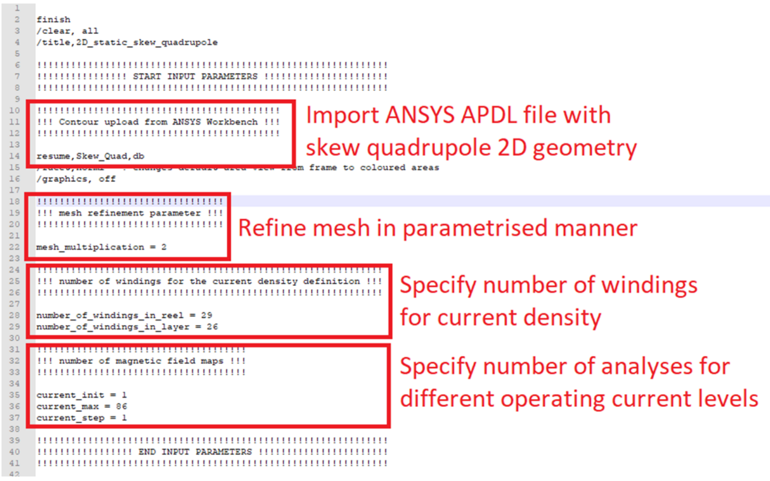

Figure 2.4: Input parameters to specify by the user in ANSYS APDL input script for the magnetic analysis.

ANSYS Workbench assigns the area numbers randomly to the created geometry in ANSYS APDL. Therefore, the user should verify the area numbers manually in ANSYS APDL and name them with the following pattern: area_yoke (iron yoke), area_coil (coil cross-section), area_airinternal (aperture side of the magnet), area_airhole (all holes in the magnet cross-section), area_airexternal (zone outside of the iron yoke). The area numbers assigned in ANSYS APDL as well as the APDL script, in which the user should name the components, respectively, are presented in Fig. 2.5.

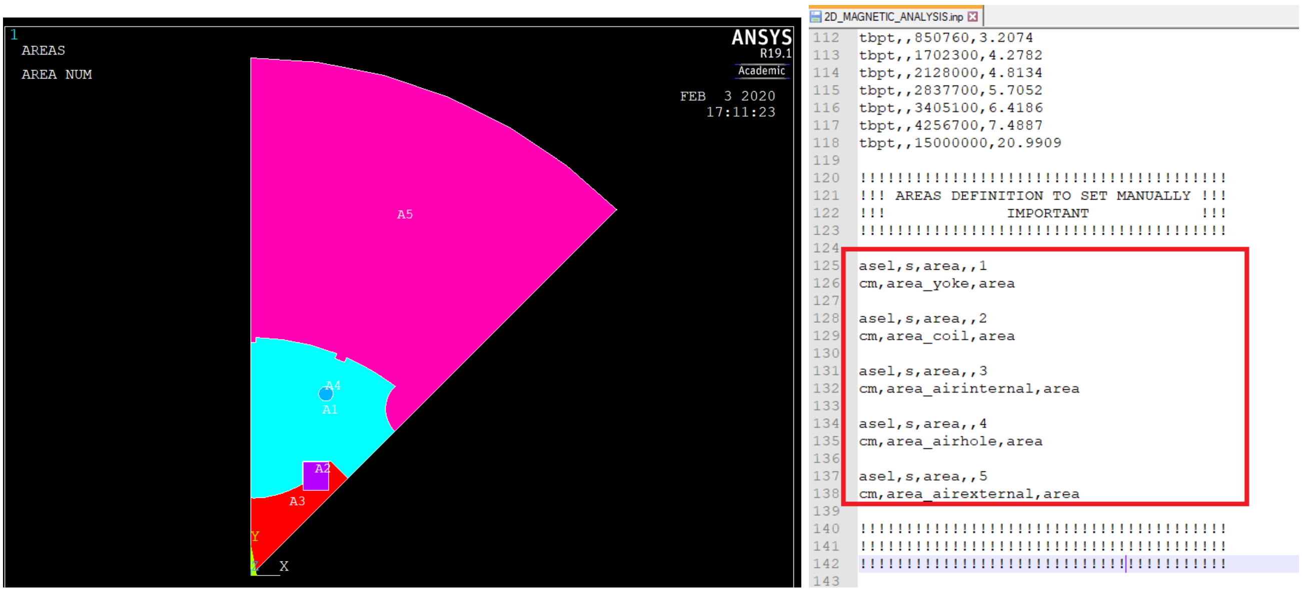

Figure 2.5: Left: areas assigned to the ANSYS APDL geometry, right: ANSYS APDL script in which the areas are assigned to the component names.

2.3 Results from ANSYS APDL

To run the magnetic simulation, the file `2D_MAGNETIC_ANALYSIS.inp` is launched in ANSYS APDL. There are seven output files from the analyses:

1. magnetic_field_current_iteration.txt
2. peak_field_coil_current_iteration.png
3. peak_field_coil_vector_current_iteration.png
4. peak_field_yoke_coil_current_iteration.png
5. peak_field_yoke_coil_vector_current_iteration.png
6. peak_field_yoke_current_iteration.png
7. peak_field_yoke_vector_current_iteration.png

The output files 2-7 only aim to visualise the simulation. The files of type 1 are important for the quench velocity-based approach, because they contain the data required to create the magnetic map as a function of current and magnetic field strength. Figure 2.6 presents the exemplary magnetic_field_current_iteration.txt file for the given value of current. In Fig. 2.6, the first two top lines of the red rectangle describe how the mesh is discretised in area_coil from Fig. 2.5 (number of mesh divisions in x- and y-direction. The third line specifies the applied current value. The red rectangle on the right side corresponds to the magnetic field strength in the Cartesian (x, y, z)-coordinate system presented in the blue rectangle.

```
Coil number of elements, x-direction = 40
Coil number of elements, y-direction = 40
Current, [A] = 1.
1 X [MM]
2 Y [MM]
3 Z [MM]
4 BX [TESLA]
5 BY [TESLA]
6 B [TESLA]
-6.9388939E-18   2.7300000E-02   0.0000000E+00   2.9725280E-04  -7.7515328E-04   8.3019385E-04
 2.4460000E-02   2.7300000E-02   0.0000000E+00   9.1531786E-05   5.3775984E-03   5.3783773E-03
 6.1150000E-04   2.7300000E-02   0.0000000E+00   8.2173522E-05  -8.6432278E-04   8.6822022E-04
 1.2230000E-03   2.7300000E-02   0.0000000E+00   3.3097816E-05  -9.2592648E-04   9.2651784E-04
 1.8345000E-03   2.7300000E-02   0.0000000E+00   1.2008511E-05  -9.6490051E-04   9.6497523E-04
 2.4460000E-03   2.7300000E-02   0.0000000E+00  -4.9338231E-06  -9.8053839E-04   9.8055080E-04
 3.0575000E-03   2.7300000E-02   0.0000000E+00  -1.6842474E-05  -9.8613424E-04   9.8627806E-04
 3.6690000E-03   2.7300000E-02   0.0000000E+00  -2.5684484E-05  -9.7896667E-04   9.7930355E-04
 4.2805000E-03   2.7300000E-02   0.0000000E+00  -3.5586139E-05  -9.6128160E-04   9.6194007E-04
 4.8920000E-03   2.7300000E-02   0.0000000E+00  -4.6432184E-05  -9.3318446E-04   9.3433890E-04
 5.5035000E-03   2.7300000E-02   0.0000000E+00  -5.5840084E-05  -8.9390056E-04   8.9564297E-04
 6.1150000E-03   2.7300000E-02   0.0000000E+00  -6.3299669E-05  -8.4257412E-04   8.4494852E-04
 6.7265000E-03   2.7300000E-02   0.0000000E+00  -7.6956655E-05  -7.8847766E-04   7.9222431E-04
 7.3380000E-03   2.7300000E-02   0.0000000E+00  -8.6785564E-05  -7.1034589E-04   7.1562771E-04
 7.9495000E-03   2.7300000E-02   0.0000000E+00  -9.7445121E-05  -6.2754490E-04   6.3506547E-04
 8.5610000E-03   2.7300000E-02   0.0000000E+00  -1.1247076E-04  -5.3118556E-04   5.4296203E-04
 9.1725000E-03   2.7300000E-02   0.0000000E+00  -1.1888564E-04  -4.2191241E-04   4.3834219E-04
 9.7840000E-03   2.7300000E-02   0.0000000E+00  -1.3272720E-04  -3.0161737E-04   3.2952929E-04
 1.0395500E-02   2.7300000E-02   0.0000000E+00  -1.4682133E-04  -1.6827338E-04   2.2332137E-04
 1.1007000E-02   2.7300000E-02   0.0000000E+00  -1.5665156E-04  -1.7372971E-05   1.5761196E-04
```

Figure 2.6: The output file from the magnetic simulation with specified data for the quench velocity-based approach.

The files magnetic_field_current_iteration.txt should be ordered from the lowest to the highest value of current, i.e. from magnetic_field_current_current_init.txt to magnetic_field_current_current_max.txt. This is carried out by default if integer values of the parameters related to operating current levels are specified (see Fig. 2.4). The files order is important for the right uploading procedure of the magnetic field strength in Python required in the following steps of the quench velocity-based approach.

Chapter 3

Quench Velocity Map Analysis

In the studies related to the quench velocity map, a set of 1D numerical analyses is conducted in ANSYS APDL with or without an additional insulation layer. The analysis is performed in ANSYS APDL with the external Python script. For more details about using the Python script in ANSYS APDL simulations, please, refer to [1, p. 51-69]. The files required to rerun the simulations are stored in: `\\cern.ch\dfs\Workspaces\a\ANSYS_Modelling\1_quench_velocity_modelling\documentation\Test_Simulation_Workflow\2_analysis_v_quench_map`.

3.1 Input json File

First of all, the user should fill the input json file to run the ANSYS APDL simulation. The default json file is placed in the folder `\input`. As presented in Fig. 3.1, the user specifies the analysis_output_directory corresponding to the directory in which ANSYS APDL is running during the simulation. In this chapter, ANSYS executes standard simulations without the quench velocity-based approach. Therefore, the user specifies the analysis_type.type as "standard".

```
"analysis_output_directory": "\\\\cern.ch\\dfs\\Workspaces\\a\\ANSYS_Modelling\\1_quench_velocity_modelling\\
                              documentation\\Test_Simulation_Workflow\\2_analysis_v_quench_map\\output",

"analysis_settings": {
    "x_quench_init": 0.0,
    "L_quench_init": 0.1,
    "t_simulation": 0.1,
    "t_com": 0.01,
    "t_step_range_min": 0.1,
    "t_step_range_max": 1.0
},

"analysis_type": {
        "type": "standard",
        "input": {
            "png_quench_state_output": true,
            "png_resistive_voltage_output": true}
    },
```

Figure 3.1: Input json file with specified analysis_output_directory and analysis_type.type marked in blue and red, respectively.

In the next step, the user specifies the data related to the current and the magnetic field strength as illustrated in Fig. 3.2. The parameters are outlined in Table 1. It is recommended to keep the value of B_initially_quenched_winding equal to B_constant.

```
"magnetic_field_settings": {
    "type": "constant",
    "input": {"B_constant": 0.0}
},

"circuit_settings": {
    "electric_ansys_elements": false,
    "electric_ansys_element_input": {"I_init": 26.0},
    "build_electric_circuit": false,
    "transient_electric_analysis": false,
    "transient_electric_analysis_input": {}
},

"temperature_settings": {
    "type": "gaussian",
    "input": {
        "T_bath": 4.3,
        "T_max": 20.0,
        "B_initially_quenched_winding": 0.0,
        "png_temperature_output": true}
    },
```

Figure 3.2: Input json file with specified values related to the magnetic field strength and the current.

Table 3.1: Parameters related to the current and the magnetic field strength.

Parameter	Type	Description
B_constant	float	Constant magnetic field strength during the analysis
I_init	float	Constant current during the analysis
transient_electric_analysis	boolean	Specifies whether the current discharge is analysed
B_initially_quenched_winding	float	Magnetic field subjected to the initially quenched winding

3.2 Running ANSYS APDL in Python

The user should launch ANSYS APDL in the Product Launcher. In order to run ANSYS APDL with Python, the given steps must be followed:

1. The ANSYS directory must correspond to the folder `\output`, as illustrated in Fig. 3.3.

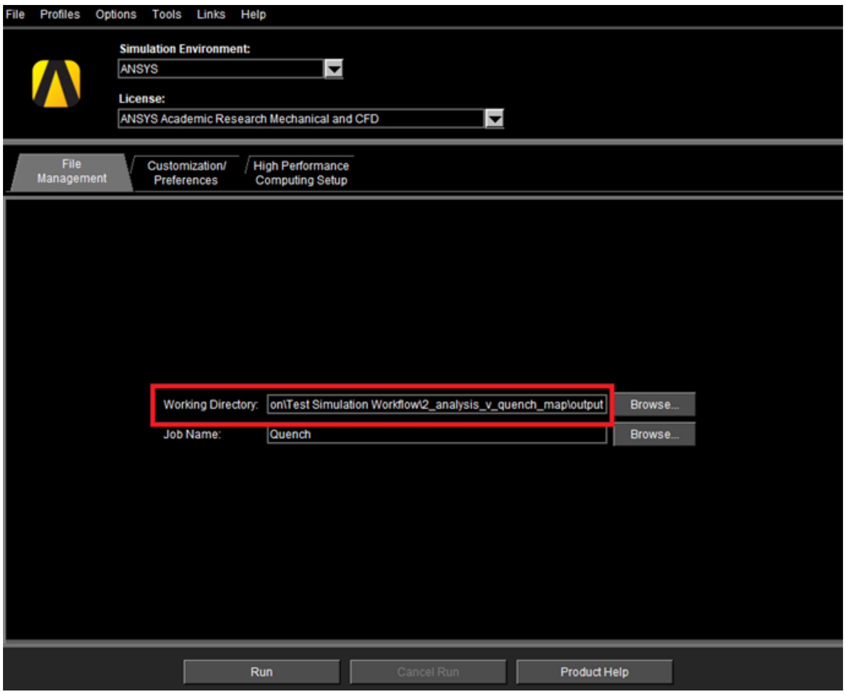

Figure 3.3: Specification of output file directory in APDL Product Launcher.

2. The additional parameter must be specified in the Product Launcher, called “-aas” as shown in Fig. 3.4. At this point, ANSYS APDL is launched by clicking the "Run" button in the Product Launcher window.

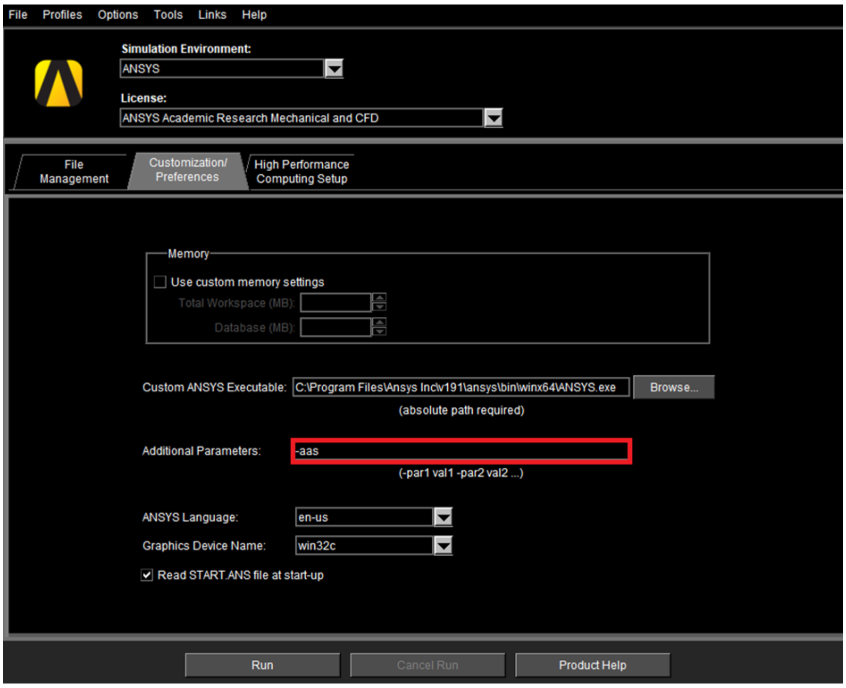

Figure 3.4: Specification of the –aas mode in APDL Product Launcher.

3. In the execution_script module of the Python script, the input_file_directory with the json filename must be specified as illustrated in Fig. 3.5.

```
# creation of analysis directories
input_files_directory = r"\\cern.ch\dfs\Workspaces\a\ANSYS_Modelling\1_quench_velocity_modelling" \
                        r"\documentation\Test_Simulation_Workflow\2_analysis_v_quench_map\input"
json_filename = "input.json"
factory = AnalysisFactory(input_files_directory, json_filename)
```

Figure 3.5: Specification of input file directory in the execution_script module of the Python script.

4. After starting ANSYS APDL, the Python script is initialised by clicking ’Run execution_script’ in PyCharm as shown in Fig. 3.6. From this point, the analysis is running until the total simulation time is computed.

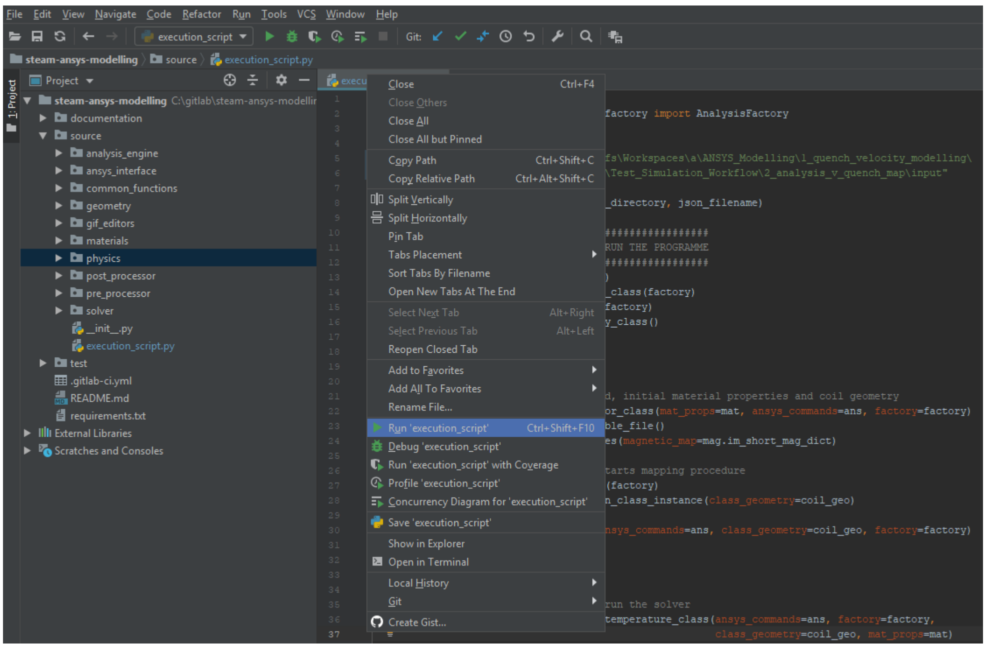

Figure 3.6: Initialisation of the execution_script in PyCharm.

3.3 Results from ANSYS APDL

The results output consists of a set of folders as shown in Fig. 3.7. The files in each of the folders are presented in Table 3.2.

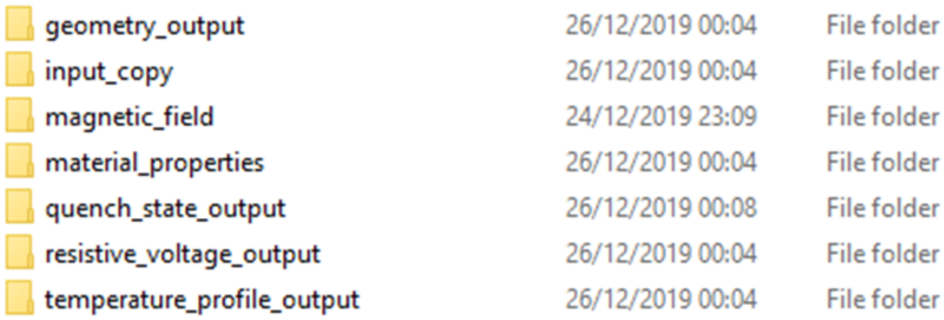

Figure 3.7: Folders in output directory created during the analysis.

Table 3.2: Description of output files from the simulation.

Folder	**File name**	**Description**
geometry_output	im_coil_length.txt	imaginary nodes vs. coil length
input_copy	input.json	input json file (copy)
magnetic_field	-	-
material_properties	item_property.txt item_property.png	property vs. temperature property vs. temperature
quench_state_output	quench_velocity.txt quench_state_iteration.txt quench_state_iteration.png	quench length, quench velocity vs. time quench state vs. coil length quench state vs. coil length
resistive_voltage_output	resistance_voltage.txt v_res_ansys.png	resistance, resistive voltage vs. time resistive voltage vs. time
temperature_profile_output	initial_energy_deposition_in_strand.txt temperature_profile_iteration.txt temperature_profile_iteration.png	initial energy in the strand temperature vs. coil length temperature vs. coil length

The file im_coil_length.txt represents a set of nodal positions in the longitudinal direction of the analysed cable. In the material_properties folder the following data are given:

1. heat capacity, thermal conductivity, thermal diffusivity of copper, Nb-Ti, and G10 and composite strand;
2. resistivity of copper and composite strand;
3. Joule heating power density profile.

The quench velocity value is stored in the quench_velocity.txt file in the folder `quench_state_output`. This value is important for the the quench velocity-based approach in further steps of the analysis. Another type of the file obtained from the simulations is Coil-Nodal-Temp_iteration.png in which the coil top view from ANSYS APDL is presented with the temperature distribution at the given iteration. The remaining files in the `\output` directory are not used for the post-processing.

Chapter 4

Minimum Propagation Zone Analysis

In the minimum propagation zone (MPZ) studies, a set of 1D numerical analyses is conducted to predict how much of initial energy is required to initialise the quench inside of a coil. For more information about the MPZ simulations, please, refer to [1, p. 85-87]. Similarly to the previous chapter, the analysis is performed in ANSYS APDL by means of the Python script. The files needed to rerun the simulations are stored in `\\cern.ch\dfs\Workspaces\a\ANSYS_Modelling\1_quench_velocity_modelling\documentation\Test_Simulation_Workflow\3_analysis_mpz`.

4.1 Input json File

The input file for the analysis is stored in the folder `\input`. In analogy to the case discussed in the previous chapter, analysis_output_directory corresponds to the directory in which ANSYS APDL is running during the simulations. As shown in Fig. 4.1, the parameter L_quench_init imposes the initial quench length in the coil. If the simulation is conducted with a symmetry condition, the value of this parameter is halved.

```
"analysis_output_directory": "\\\\cern.ch\\dfs\\Workspaces\\a\\ANSYS_Modelling\\1_quench_velocity_modelling\\
                              documentation\\Test_Simulation_Workflow\\3_analysis_mpz\\output",

"analysis_settings": {
    "x_quench_init": 0.0,
    "L_quench_init": 0.2,
    "t_simulation": 0.1,
    "t_com": 0.01,
    "t_step_range_min": 0.1,
    "t_step_range_max": 1.0
},
```

Figure 4.1: Input json file with specified analysis_output_directory and L_quench marked in blue and red, respectively.

Figure 4.2 presents the input parameters corresponding to the initial temperature profile. They are described in Table 4.1.

```
"temperature_settings": {
    "type": "gaussian",
    "input": {
        "T_bath": 4.3,
        "T_max": 20.0,
        "B_initially_quenched_winding": 0.0,
        "png_temperature_output": true}
    },
```

Figure 4.2: Input json file with specified values related to the initial temperature profile.

Table 4.1: Parameters related to the current and the magnetic field strength.

Parameter	**Type**	**Description**
T_bath	float	Initial bath temperature
T_max	float	Maximum temperature of the Gaussian profile
B_initially_quenched_winding	float	Magnetic field which serves for calculating the critical temperature to estimate the quench front position in the Gaussian profile
png_temperature_output	boolean	Specifies whether the output temperature profile is plotted as the simulation continues

4.2 Running ANSYS APDL in Python

The case corresponds to that, presented in Section 3.2.

4.3 Results from ANSYS APDL

The case corresponds to that, presented in Section 3.3.

Chapter 5

Quench Analysis of Skew Quadrupole

This chapter an exemplary analysis of the skew quadrupole with the quench velocity-based approach. The analysis can be used for the quench simulation of any high-order corrector geometry. For further information concerning the quench analysis of the skew quadrupole, please, refer to [1, p. 70-97]. The files required to rerun the exemplary simulation are stored in the following directory: `\\cern.ch\dfs\Workspaces\a\ANSYS_Modelling\1_quench_velocity_modelling\documentation\Test_Simulation_Workflow\4_analysis_high_order_corrector`.

5.1 Input Files

As presented in Fig. 5.1, in order to run the simulation, the Python script requires a set of input files such as inductance.txt, quench_velocity_map.txt, and input.json. The file inductance.txt represents the evolution of the magnet inductance as a function of current. The file quench_velocity_map.txt is based on the simulations discussed in Chapter 3. The input.json file is a text file with a set of simulation parameters similarly to the analyses discussed in the previous chapters. The folder `\magnetic_field_map` stores a set of magnetic field maps for a varying value of the operating current. The files are obtained from the magnetic simulations discussed in Chapter 2.

magnetic_field_map	07/02/2020 14:32	File folder
inductance.txt	15/12/2019 16:10	Text Document
input.json	07/02/2020 16:49	JSON File
quench_velocity_map.txt	26/12/2019 00:12	Text Document

Figure 5.1: Input files required to run the simulation.

5.2 Input json File

Apart from the data discussed in the previous chapters, the quench velocity-based approach requires additional information concerning the quench velocity map as a function of current and magnetic field strength. The input.json file contains the data in the analysis_type part of the text file as illustrated in Fig. 5.2. The parameters are described in Table 5.1.

```
"analysis_type": {
        "type": "v_quench_based_approach",
        "input": {
            "v_quench_model": "numerical",
            "v_quench_map_filename": "quench_velocity_map.txt",
            "png_quench_state_output": true,
            "png_resistive_voltage_output": true}
    },
```

Figure 5.2: Input json file with specified values related to the quench velocity-based approach.

Table 5.1: Parameters related to the quench velocity-based approach.

Parameter	Type	Description
v_quench_model	string	Specifies what type of quench velocity map is chosen
v_quench_map_filename	string	Specifies filename with the quench velocity map
png_quench_state_output	boolean	Specifies whether the quench state profile is plotted as the simulation continues
png_resistive_voltage_output	boolean	Specifies whether the resistive voltage profile is plotted as the simulation continues

Figure 5.3 depicts the analysis settings corresponding to the magnetic field strength as well as the circuit. The parameters are described in Table 5.2. The parameters used for the discharge of the skew quadrupole are explained in [1, p.70-76 and 91].

```
"magnetic_field_settings": {
    "type": "2D_static",
    "input": {
        "B_map_foldername": "magnetic_field_map"}
},

"circuit_settings": {
    "electric_ansys_elements": true,
    "electric_ansys_element_input": {"I_init": 86.0},

    "build_electric_circuit": true,
    "transient_electric_analysis": true,
    "transient_electric_analysis_input": {
        "R_dump": 2.0,
        "L_diff_filename": "inductance.txt",
        "V_th": 0.05,
        "t_delay": 0.005,
        "I_discharge_criterion": 80.0}
        },
```

Figure 5.3: Input json file with specified values related to the circuit settings and the magnetic field mapping procedure.

Table 5.2: Parameters related to the circuit settings and the magnetic field mapping procedure.

Parameter	Type	Description
B_map_foldername	string	Foldername in which the magnetic field map is stored (see Fig. 5.1)
electric_ansys_elements	boolean	Specifies whether LINK68 elements are used to model the composite strand (default for quench velocity-based approach)
I_init	float	Initial current in the analysis
build_electric_circuit	boolean	Specifies whether the independent current source, dump resistor, and inductor are built in ANSYS APDL
transient_electric_analysis	boolean	Specifies whether the magnet discharge is simulated
R_dump	float	Dump resistor value added to the circuit
L_diff_filename	string	Filename of the differential inductance as a function of current (see Fig. 5.1)
V_th	float	Threshold voltage value at which the quench is detected
t_delay	float	Time after which the current source is disconnected from the circuit after reaching the threshold voltage
I_discharge_criterion	float	Current value at which the simulation ends during the discharge

The high-order corrector geometry parameters are specified in the input json file as well. They are shown in Fig. 5.4 and further described in Table 5.3. The geometrical meaning of the parameters is discussed in [1, p.81-83 and 87-88].

```json
"geometry_settings": {
    "dimensionality": "multiple_1D",
    "type": "high_order_corrector",
    "type_input": {
        "d_strand": 0.7,
        "d_ins": 0.84,
        "a_strand": 0.941,
        "u_ins": 1.0,
        "u_resin": 0.5,
        "e_coil": 387.3,
        "d_coil": 82.9,
        "rad_coil": 9.15,
        "n_windings": 9,
        "n_layers": 3,
        "n_turns_in_layer": 3,
        "which_layer_first_in_analysis": 1,
        "which_turn_first_in_analysis": 1,
        "type_insulation_settings": {
            "insulation_analysis": true,
            "insulation_analysis_input": {"n_divisions_ins": 2}
        },
        "type_mesh_settings": {
            "n_divisions_e_coil": 26,
            "n_divisions_d_coil": 6,
            "n_divisions_rad": 2
        }
    }
}
```

Figure 5.4: Input json file with specified values related to the geometry of the high-order corrector.

Table 5.3: Parameters related to the geometry of the high-order corrector.

Parameter	Type	Description
d_strand	float	Diameter of the composite strand
d_ins	float	Diameter of the composite strand with insulation
a_strand	float	Side of the square resin area assigned to one composite strand
u_ins	float	Coefficient of the insulation element area
u_resin	float	Resin packing coefficient
e_coil	float	Length of the coil long side
d_coil	float	Length of the coil short side
rad_coil	float	Initial radius of the coil between its long and short side
n_windings	int	Total number of analysed windings
n_layers	int	Total number of analysed layers
n_turns_in_layer	float	Total number of turns in a single layer
which_layer_first_in_analysis	int	Layer number from which the geometry is created
insulation_analysis	boolean	Specifies whether the insulation layer with turn-to-turn propagation is analysed
n_divisions_ins	int	Number of mesh divisions across the insulation elements
n_divisions_e_coil	int	Number of mesh divisions along the long side of the coil
n_divisions_d_coil	int	Number of mesh divisions along the short side of the coil
n_divisions_rad	int	Number of mesh divisions along the each of curvatures of the coil

5.3 Running ANSYS APDL in Python

The case corresponds to that, presented in Section 3.2.

5.4 Results from ANSYS APDL

The files obtained from the analysis with the quench velocity-based approach are similar to that presented in Section 3.3. The difference is based on two file types described in Table 5.4. The files marked in red represent those that are not included in the output folders with respect to the case described in Section 3.3. First of all, the file magnetic_map.png is obtained from the simulation that is a graphical representation of the magnetic field strength assigned to each winding of the simulated coil. Second of all, the file quench_velocity.txt is not included in the results due to the fact that the quench velocity is an input parameter in the quench velocity-based approach.

Table 5.4: Description of additional output files from the simulation.

Folder	**File name**	**Description**
magnetic_field	magnetic_map.png	magnetic field strength vs. winding position across the coil
quench_state_output	quench_velocity.txt	quench length, quench velocity vs. time

In addition, there are the following output files in the directory `output`:

1. Coil-Nodal-Temp_iteration.png with the the coil top view from ANSYS APDL showing the temperature distribution at the given iteration;
2. Coil-Nodal-Volt_iteration.png with the the coil top view from ANSYS APDL showing the voltage distribution at the given iteration;
3. sol_dump_resistor.inp with the values of current, voltage and power in the resistor at the given time step;
4. sol_inductor.inp the values of current, voltage and power in the inductor at the given time step.

Chapter 6

HPC Cluster Implementation

During the last month of the project (February 2020), an effort was made to run the ANSYS simulations on the CERN cluster. This chapter presents how the standard ANSYS simulation without the external Python script runs on the cluster. The simulations with the quench velocity-based approach were finally not launched due to the lack of time. The following sections describe the preliminary steps required to start the standard ANSYS simulation on the cluster. In addition, some recommendations to future software developers on how to include the quench velocity-based approach in the cluster are mentioned. The files required to rerun the exemplary simulation on the cluster are stored in the given directory: `\\cern.ch\dfs\Workspaces\a\ANSYS_Modelling\1_quench_velocity_modelling\documentation\Test_Simulation_Workflow\5_hpc_normal_ansys_analysis`.

6.1 Further information about the cluster at CERN

It is recommended to launch the ANSYS APDL simulations on the HTCondor cluster. For more information, please refer to the CERN documentation presented in `https://cern.service-now.com/service-portal/article.do?n=KB0006080`. A set of lectures is additionally dedicated to the cluster users at CERN accessible under the following link: `https://indico.cern.ch/event/811481/`.

6.2 Simulation Prerequisites

Before running the simulation, the user should follow the given steps:

1. install PuTTY software from CERN Computer Framework (CMF);
2. create a folder for the cluster analyses on the personal CERNBox account;
3. create the AFS repository.

The AFS repository serves for submitting the APDL and htcondor files to run the ANSYS simulation on the cluster. The AFS configuration is performed as follows:

1. go to the CERN Resource Portal at `https://resources.web.cern.ch/resources/Manage/ListServices.aspx`;
2. click on "List Services";
3. click on "LXPLUS and Linux" in the "Operating Systems" section;
4. click on the "Computing Groups" blue tab on the left;
5. click on the "Computing Groups" blue tab on the left;
6. click on the "Subscribe to AFS" blue button in the "Manage your AFS subscription" section and the home directory will be created (note that it may take a few 2 hours);
7. contact the CERN service desk for the public key configuration given in `https://cern.service-now.com/service-portal/article.do?n=KB0006086`.

6.3 Submitting the ANSYS script to the cluster

The simulation is submitted from AFS by typing a set of commands in PuTTY as described in `https://cern.service-now.com/service-portal/article.do?n=KB0006082`. Based on the files stored in `\cluster_files`, the user should follow the given steps:

1. copy the files `my_htcondor.sub`, `my_script.sh`, and `test_simulation.txt` to the CERN-Box;
2. move the files `my_htcondor.sub` and `my_script.sh` to AFS (*mv* command in PuTTY);
3. create three directories in AFS: output, error, log (*mkdir* command);
4. submit the file `my_htcondor.sub` in PuTTY (*condor_submit* command).

The user should make sure that `my_script.sh` has the correct Linux/Unix text format because new lines in Windows text files are implemented in a different manner (*dos2unix* command). After the correct simulation submission to the cluster, the analysis is computed and the results from the simulation are sent back to the user's personal CERNBox. The output files from the simulation performed on the cluster should correspond to those stored in the folder `\output`.

6.4 Recommendation of future code development

The analysis with the quench velocity-based approach could not be launched because it is not possible to run the ANSYS Product Launcher on the cluster as in case of a personal computer usage presented in Section 3.2. For future cluster applications, it is recommended to extend the Python code with the script able launch the ANSYS APDL script without the Product Launcher. In such a case, an additional information would be required in the input json file about the directory in which the ANSYS software is stored on the personal computer or the cluster. Moreover, the Python package given in `https://gitlab.cern.ch/steam/steam-ansys-modelling` would have to be imported to the cluster.

Before implementing those steps (which are only a recommendation), the CERN service desk should be contacted.

Chapter 7

General Comments on the Developed Simulation Tool

The development of the quench simulation tool lasted one year and there are still many options to improve its performance. In two following sections, the process of choosing ANSYS APDL as the simulation tool, as well as the proposals for the future software development are presented.

7.1 Choice of the simulation tool from the ANSYS package

ANSYS package consists of multiple tools able to solve thermal problems including ANSYS APDL, ANSYS Thermal, ANSYS CFX, or ANSYS Fluent. ANSYS APDL is a separate module able to solve multi-physics problems related to, among others, mechanical, thermal, or magnetic studies. The remaining ANSYS tools correspond to the separate GUI modules in ANSYS Workbench. Table 7.1 presents the advantages and disadvantages of implementing the quench velocity-based approach in ANSYS APDL or in any module of ANSYS Workbench. The main reason for choosing ANSYS APDL was the possibility of connecting it to an external code in which the quench velocity-based approach was implemented. The solution to this problem was not found in case of ANSYS Workbench.

Table 7.1: Advantages and disadvantages of the proposed tools.

Simulation tool	**+ Advantages**	**- Disadvantages**
ANSYS APDL	• It is possible to connect an external Python script able to co-simulate an external routine with the numerical ANSYS solver. • ANSYS Parametric Design Language (APDL) is used to define and run the simulation. APDL scripts are independent of new ANSYS versions which allows for an easy archiving process of the project. • APDL scripting allows to parameterise the simulation (no "clicking" in GUI).	• Python implementation in ANSYS APDL tool is based on an external library ansys_corba which is not supported by the provider of ANSYS software. Therefore, a part of the Python code responsible for communicating with ANSYS APDL is version-dependent. • Advanced knowledge of APDL is required (large documentation is provided by ANSYS). • ANSYS support in APDL scripting is limited due to the time-consuming bug-fixing process.
ANSYS Workbench	• Basic/no knowledge of APDL is required. • ANSYS support is much more responsive because the bugs occurring in Workbench are less customised. • Simplified procedure to use a cluster based on the RSM module in ANSYS.	• Implementation of additional scripts responsible for the co-simulation of an externally developed routine remains a difficult task (a customised problem in ANSYS Workbench leads to the limited ANSYS support)

7.2 Possible future development

Figure 7.1 illustrates the possible future code code development as a function of the computation time with respect to the necessary workload. The proposals are numbered in the order of the recommended implementation.

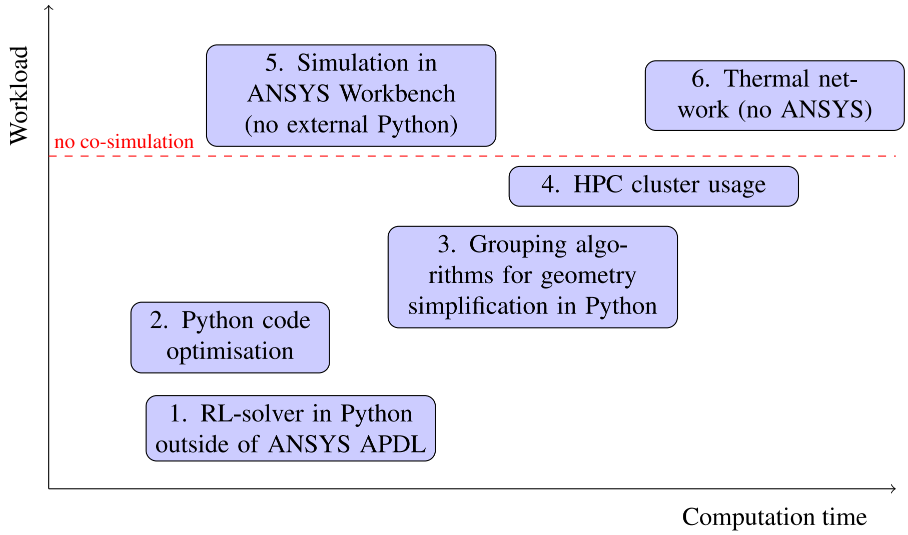

Figure 7.1: Possible project development directions as a function of the workload with respect to the gain in the computation time.

Based on the numbering order shown in Fig. 7.1, each of the proposed code development is further explained below:

1. At this moment, ANSYS APDL solves both the thermal coil domain and the electric discharge of an RL-circuit (please, refer to [1, p. 76]). The strong coupling is implemented between the thermal coil domain and the RL-circuit, being the only available option in ANSYS APDL. Since the RL-circuit requires a lower time step to reach the desired precision, the quench velocity-based approach related to the thermal coil domain with less strict time step requirements is slowed down. The solution to this problem would be the preparation of an electrical numerical solver in Python simulating the RL-circuit discharge. The circuit would be connected to ANSYS APDL in which only the coil would be solved. The solvers in ANSYS APDL and Python could be connected in form of a weak coupling.
2. It is recommended to verify the Python code in view of its computation efficiency. The code can be optimised in order to reduce the number of operations required to achieve the goals of the quench velocity-based approach explained in [1].
3. The grouping algorithms would aim to reduce the number of windings modelled in the geometry by assuming that a set of windings quenches at the same time. As presented in Fig. 7.2, each red rectangle is modelled as one winding which allows for reducing the number of degrees of freedom required to discretise the coil.

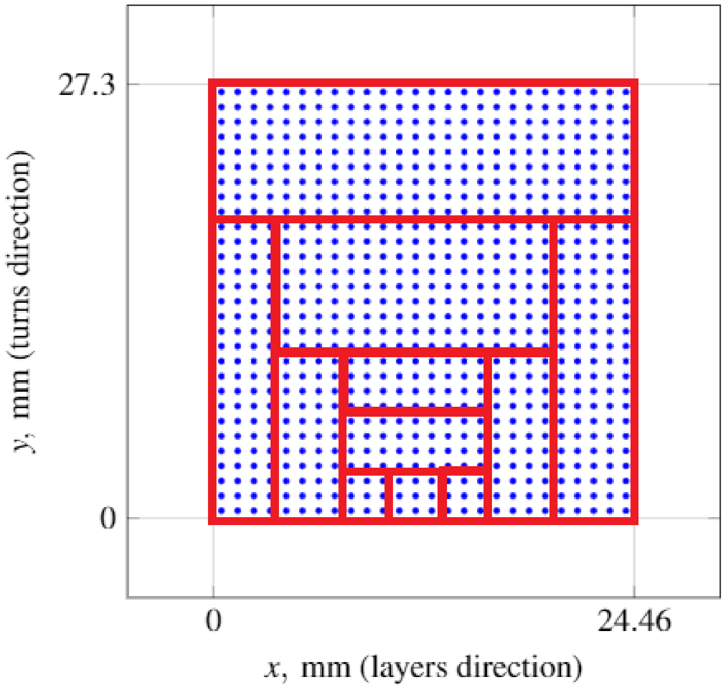

Figure 7.2: Schematic of the grouping algorithm in the coil geometry (based on the skew quadrupole).

4. As discussed in Chapter 6, the CERN cluster usage is recommended to shorten the computation time of the quench simulations with the quench velocity-based approach.
5. The implementation of the entire quench velocity-based approach in the ANSYS tool would allow for shortening the computation time by excluding the data exchange between ANSYS and the Python script.
6. Similarly to the previous case, the preparation of the thermal solver outside of ANSYS would allow shortening the computation time related to the data exchange between ANSYS and the Python script. Moreover, the customised numerical solver could possibly be faster than the commercial ANSYS tool.

Bibliography

[1] M. Wilczek, “Finite element analysis of magneto-thermal transient effects in superconducting accelerator magnets,” Master’s thesis, Lodz University of Technology, January 2020.