

Файловая структура проекта

Корень проекта – так в дальнейшем будет называться директория *MagneticField*

MagneticField/doc – директория с *PDF*-файлами, в которых содержатся документация к двум версиям *magPyLib* и несколько компиляций изображений, относящихся к нескольким исследованиям структуры магнитного поля.

MagneticField/pic – директория с рисунками, использовавшимися при составлении отчета, внутри которой расположена директория

MagneticField/py – директория с пайтон-файлами для проекта с *magPyLib*, подробнее описывается ниже

MagneticField/bash-scripts – директория с вспомогательными скриптами к проекту *magPyLib*, подробнее описывается ниже

MagneticField/config-circle.yaml – конфигурационный ямл-файл для проекта с *magPyLib*, подробнее описывается ниже

MagneticField/config-rect.yaml – конфигурационный ямл-файл для проекта с *magPyLib*, подробнее описывается ниже

MagneticField/openEMS – директория для проекта с *openEMS*

Проект модели катушек с использованием библиотеки *magPyLib*

MagneticField/py – содержит следующее:

- 1) *main-dist.py* – файл, запускающий моделирование электромагнитного поля катушек с постоянным сигналом, содержащий внутри себя вспомогательный цикл, итеративно увеличивающий расстояние между катушками для изучения структуры магнитного поля. Первым делом пользователю нужно поменять значение переменной *configPath* так, чтобы путь к конфигурационному файлу был ликвиден относительно его файловой системы. В случае, если пользователю нужна круглая катушка, то ему также необходимо изменить название файла на *config-circle.yaml*, а также изменить фрагменты инициации начальных

значений для соответствия со структурой конфигурационного файла. В случае, если все было сделано правильно, то можно запустить программу командой *python main-dist.py*, предварительно установив библиотеку *magPyLib* командой *pip install magpylib*. При удачном запуске в директории *MagneticField/pic/fields/XY* появятся .png файлы, соответствующие каждому сечению магнитного поля при соответствующем расстоянии между катушками. Также проект позволяет смотреть поле в более чем одной секущей плоскости, а кроме того, в параллельных ей плоскостях тоже. Для этого нужно раскомментировать часть значения переменной *planeNames*, а также изменить переменную *zonesNumber* в конфигурационном файле. При изменении размеров катушек и дистанции между ними через конфигурационный файл, следует обращать внимание, чтобы секущая плоскость продолжала их охватывать, возможно, пользователю понадобится поменять значение инкремента расстояния каждой катушки от центра координат. Также пользователю может быть интересно несколько изменить структуру генерируемых изображений (например – подписи к осям и рисункам). Для этого следует перейти к последнему фрагменту *main-* файла и несколько модернизировать структуру вызовов некоторых функций библиотеки *matplotlib*.

- 2) *main-dist.py* – файл для моделирования магнитного поля катушек с постоянной шириной прохода, но питающихся от псевдосинусоидальных сигналов различных частот. По умолчанию оба конфигурационных файла содержат параметры синусов, но в случае с использованием *main-dist.py* они просто не используются. Дискретизация синусов выбрана таким образом, чтобы оба сигнала рассматривались на *periodsNumber* числе периодов сигнала с наименьшей частотой (следовательно, тот у которого частота выше уложится в этот интервал более чем полностью) и число рассматриваемых дискретных значений синусов выбирается параметром *stepsNumber* относительно периода сигнала с наибольшей частотой (следовательно, сигнал с меньшей оцифруется еще более точно). Если пользователю понадобится смотреть только одну катушку

(или больше чем две), то необходимо отредактировать раздел с созданием катушек (в соответствии с требуемым числом экземпляров катушек), объединение экземпляров в одну коллекцию, а также – конфигурационный файл.

- 3) *src/Plane.py*, *src/CoilLoops.py* и *src/coilRect.py* являются классами, используемыми в мейн-файлах и представляют абстракции текущих плоскостей, кругловитковых катушек или катушек с прямоугольными витками соответственно. Предполагается, что пользователю их модифицировать не нужно.
- 4) *bash-scripts/make-pdf.sh* и *bash-scripts/clear.sh* – вспомогательные баш-скрипты, предназначенные для того, чтобы объединять в совокупные pdf-файлы изображения полей в текущих плоскостях, а также очищать директории *pic/field/XY*, *pic/field/XZ* и *pic/field/YZ* от изображений и pdf. Запускать скрипты следует непосредственно из *bash-scripts/*, предварительно снабдив их правами на исполнение.
- 5) *main-test.py* представляет собой пример необходимых изменений для переключения с прямоугольных катушек на круглые.

Использование проекта предполагает, что пользователь внесет минимально необходимые изменения в мейн-файлы, описанные выше, а после будет лишь подстраивать значения параметров в конфигурационных файлах.

Проект модели катушек с использованием *openEMS*

Несмотря на то, что желаемые частоты в программе *openEMS* не дадут ликвидного результата при таких малых расстояниях, на которых изучается поле, была создана простая модель двух прямоугольных катушек, питаемых источниками с частотой в 1 МГц. Результат ее работы можно увидеть в отчете за 11-е августа. *Matlab/Octave*-скрипт, запускающий моделирование – *test-coil.m* располагается в директории *MagneticField/openEMS/*. Перед его использованием необходимо установить *octave* как оболочку для написания таких скриптов, *paraview* для визуализации результатов моделирования и собрать непосредственно *openEMS*, в соответствии с инструкциями, с официального сайта *openEMS*: <https://openems.de/start/>.

Полезными при работе с *openEMS* оказались следующие статьи (которые коротко и доступно объясняют принципы в соответствии с которыми был создан скрипт): <https://habr.com/ru/post/255317/>, <https://habr.com/ru/post/258489/> и <https://habr.com/ru/post/259383/>, а также следующие ссылки с официального сайта: <https://openems.de/index.php/Tutorials> и https://openems.de/index.php/List_of_Functions.html.