9. Data Management Plan

Our proposed project involves open-source computational electromagnetism (CEM) software development, and the creation of a mobile application. Our data management plan (DMP) is simple and straightforward. We will maintain an open-source philosophy, ensuring that users can access our CEM codes, schematic files, and output data. We will also ensure that our application code is open-source, while protecting student data. We share below the six data products we anticipate creating through our project, and elaborate our data management given NSF and Engineering Directorate guidelines. We describe the data products and the standard formats we will use. We share how we will ensure appropriate access to our data products, and we share our vision for the use of our data products. Finally, we explain how we will archive the data in a way that preserves its utility for future research.

Our proposed project will produce six data products. (1) Python3 code and Jupyter notebooks, (2) schematics in GDSII format, converted to STL files for open-source CAD and 3D printing, (3) HDF5 output files from CEM calculations, (4) RF testing and calibration data, (5) Jupyter notebooks for educational and curricular purposes, and (6) the EASTLOS Android application.

Our Python3 code will use the open-source MEEP and scikit-learn packages, distributed under the GNU GPLv3 and BSD licenses, respectively. Depending on the use case, we will create this code in the form of scripts or Jupyter notebooks. MEEP scripts have an enormous advantage for CEM, in that they can be run with parallel processing via MPI. It is often helpful, however, to store well-documented examples in Jupyter notebooks. Thus, the majority of our code will be in script form, accompanied as appropriate with demonstrations in Jupyter notebooks. Jupyter notebooks can be extensively formatted for web-based usage with the Markdown language. Thus, important metadata required to run our code will be shared via the notebooks. Both notebook files (.ipynb) and script files (.py) will be stored in an official, public GitHub repository. Users will be able to access output data in the Poet Commons. The Poet Commons is a web-accessible long-term archive for members of the Whittier College community to preserve academic work. When our results are mature, we will make our Poet Commons archive visible to the internet.

The GDSII file format is a binary database which is the industry standard for Electronic Design Automation (EDA) data exchange of integrated circuit (IC) layouts. One use of GDSII (.gds) files is the transfer of layouts between IC design tools. We have adapted the GDSII file format to RF antenna design. GDSII designs can be imported into MEEP, which serves as our CEM engine for RF design. Given the standard nature of the GDSII file format, it is natural to include GDSII design files alongside our Python3 and Jupyter notebooks in the GitHub repo and Poet Commons archive. Users should be able to access our CAD designs so that our codes can be run in the same way we run them. Individual GDSII files have a maximum size of 2^{16} bytes, or 65kB. Thus, they stay under the maximum file size for our public GitHub repository.

Using an open-source codes available on GitHub, we can extrude two-dimensional GDSII layers into 3D "stereolithography" STL files. STL files are one standard used for additive manufacturing. Most 3D printing platforms can accept STL files as inputs. It is sensible to store the STL files we generate, rather than assuming users can perform the GDSII to STL translation on their own. The open nature of the STL format also ensures our project results can be printed by others in an open and transparent way. Thus, we commit to storing our STL files alongside the GDSII, Jupyter notebook, and Python3 files in our open GitHub repository.

Our CEM and machine learning codes will generate results in the HDF5 format. Users will want to check their results against ours. The mission of the HDF Group is to "ensure efficient and equitable access to science and engineering data across platforms and environments, now and forever." We agree, but we have to specify *how* we will ensure access. HDF5 output can exceed git file-size limits. Fortunately, there is precedent at Whittier College for sharing access to scientific data via the Whittier Domains platform. For example, G. Piner and P. G. Edwards have

organized a catalogue of astrophysical TeV-blazar data as a Whittier Domains portfolio. Thus, Whittier Domains should be the appropriate venue to host our HDF5 file outputs. When our project is mature, we can link the portfolio to the long-term Poet Commons archive.

Our designs will be tested in the laboratory, and the testing data the fifth data product associated with the proposed work. Given the limted size of this data product, it makes sense to store it in our GitHub repository alongside the other data types. The only limiting factor is file size. Our publication practice for our most recent publications sharing this data type has been to use git to track our LaTeX article templates. Our RF testing data will be difficult to interpret without the corresponding metadata, so we commit to including all necessary metadata. Thus, our complete vision for the GitHub repository contains all code, analysis scripts, design files, and journal article writing, in a common location. Colleagues should be able to reproduce our results using the resources and data products in the repository. Our products on GitHub will always be backed up locally at Whittier College, and HDF5 data output will reside in Whittier Domains and the Poet Commons.

The educational Jupyter notebooks associated with the course integration portion of our proposal are most useful to the instructors of Whittier College courses. The most natural way to share these products with our colleagues is Moodle, our content management system (CMS). The Whittier College Moodle, however, requires access granted by the institution. Whittier Domains and Poet Commons, however, do not. Thus, Whittier Domains and the final Poet Commons archive are the appropriate locations to facilitate long-term sharing of learning modules that utilize the results of our research.

A common advantage of GitHub, Whittier Domains, and Poet Commons is that our group does not have to maintain these sites. We can manage our portfolios and repositories on these platforms periodically, respond to data sharing requests by sharing links to them, and disseminate knowledge of them through our publications. We commit to citing these repositories in our future publications, as we have done in the past. One example of this practice is "Complex analysis of Askaryan radiation: A fully analytic treatment including the LPM effect and Cascade Form Factor," (Hanson, J.C. and Connolly, A.L., Astroparticle Physics, 2017). In this publication, we created a C++ code that contained an analytic model we produced and studied. Users were directed to the code repository from within the publication.

The main intellectual focus of our project is to create an open-source ecosystem for RF design and production through additive manufacturing. Thus, we will ensure others can use our data products to the fullest extent. Our open distribution policy is informed by this. We will ensure that all data products are accompanied by instructions and metadata necessary to reproduce our results. From within our publications, we can provide the details of successful additive manufacturing procedures. We will publish our results in the open access journal *Electronics* (MDPI) in order to facilitate sharing these details. Open access journals are especially important for HSIs like Whittier College that cannot devote the resources to subscribe to a wide variety of specialty engineering journals.

Our proposal also involves the creation of an Android mobile application. The goal is to boost student engagement, enthusiasm, and learning in introductory STEM courses. Students can generate data from within the app that could reveal private information, and we recognize that it is our responsibility to protect our students. Thus, we will only consider sharing app data with research colleagues on a case-by-case basis. We see no reason that the application source code cannot be included in our public GitHub repository under the same BSD license as scikit-learn. If the application succeeds in boosting undergraduate enagement, sharing it is in the best interest of the community.