other PACs numbers?

Eqtools: Modular, Extensible, Open-Source, Cross-Machine Python Tools for Working with Magnetic Equilibria

M. A. Chilenski,[*] I. C. Faust, and J. R. Walk

MIT Plasma Science and Fusion Center

(Dated: November 10, 2014)

An introduction to the eqtools package, a modular, extensible, open-source toolkit in the Python programming language for handling magnetic equilibria from tokamaks, is presented. The eqtools package provides a single interface for working with magnetic equilibrium data, both for handling derived-quantity data and mapping between coordinate systems in the flux grid, extensible to function with data from different experiments, data formats, and magnetic-reconstruction codes, replacing the static solutions currently used on tokamak experiments. Moreover, the development of magnetic-equilibrium functionality in the Python language removes a substantial barrier to code migration and new development in Python, which presents a number of attractive advantages. In this paper, we introduce the modular structure and design of the eqtools package and detail the workflow for usage and expansion to additional devices. The implementation of a novel three-dimensional spline solution (in two spatial coordinates and in time) for improved timebase accuracy is also detailed. Finally, Benchmarking for accuracy and speed against existing methods are detailed.

PACS numbers: 52.55.Fa

[*] markchil@psfc.mit.edu

I. INTRODUCTION

The basic computational tasks associated with magnetic-equilibrium reconstructions – namely, the handling of derived quantities (*e.g.,* calculated plasma current, safety-factor profiles) and the mapping between real-space and flux coordinate systems for experimental data – are universal among tokamak experiments. Despite this commonality, experiments typically utilize in-house solutions developed for the particulars of that experiment's data storage and usage. This ad-hoc development of base-level functionality inhibits the mobility of higher-level codes to other devices (as the code may require substantial modification to address the particulars of the new data storage design). Moreover, such development is often quite static, such that the implementation is difficult to extend to new data formats (for example, to handle both a primary MDSplus-based data storage system cite? and the *eqdsk* storage files produced directly by the EFIT reconstruction code cite), necessitating parallel workflows for functionally identical tasks depending on the data source. Best design practices call for the vagaries of data source and storage implementation to be placed in the back end, presenting a consistent, straightforward interface common between machines and code implementations to the research scientist.

ACKNOWLEDGMENTS

acknowledgements go here.