

SAMPL: Simple Accelerator Modelling Python Library

T. Price

Version 1.0: January 12, 2018

Abstract

SAMPL has been created from the simulation code SAMM (Simple Accelerator Modelling in MatLab) written by Andy Wolski, for the purposes of having simulation code to be used to investigate trends in the machine VELA-CLARA. These notes describe the equations behind each section of code in SAMPL that maps particles through a given element used in an accelerator.

Contents

1	Introduction	3
1.1	Introduction to SAMM	3
1.2	Introduction to SAMM	3
2	Drift	3
3	Solenoid	3
4	Dipole	4
5	Quadrupole	4
6	Sextupole	4
7	Octupole	4
8	Multipole	4
9	RF Cavity	4
10	RF Accelerating Structure	5
11	NEW: Solenoid and RF Class	5
12	Orbit Corrector	5
13	Screen	5
14	Beam Position Monitor	5

1 Introduction

In this section, we give a brief explanation of the purpose and capabilities of SAMM, describe how to install it, and discuss some simple examples.

1.1 Introduction to SAMM

SAMM is a set of Matlab [1] routines for modelling beam dynamics in high energy particle accelerators. To develop an application, the user writes Matlab scripts that call classes and functions defined in SAMM. In other words, SAMM consists of a library from which an accelerator simulation may be developed. The intentions behind SAMM are: for the physics to be as accurate as possible; for the code to be as short and simple as possible; and for the structure of the code to reflect as closely as possible the structure of a real accelerator. It is also intended that the code be structured to make it as easy as possible to extend its capabilities: that is, it should be straightforward for the user to add new beamline components, dynamical effects, and analysis procedures, as required.

Since simplicity and brevity are prioritised over performance, SAMM is unlikely to be the best code for large-scale simulations of complex accelerator systems. However, some effort has been made to make the code reasonably efficient, so for relatively simple systems, the speed of computation should be acceptable.

To achieve a logical and clear structure in SAMM, extensive use is made of object-oriented programming features in Matlab. Users are recommended to familiarise themselves with the section on “Object-Oriented Programming” in the Matlab documentation [?], before working with SAMM.

The latest version of SAMM has been tested with Matlab Release 2012a. Compatibility with earlier versions of Matlab is not guaranteed.

1.2 Introduction to SAMM

12

2 Drift

zdfhzg

3 Solenoid

SAMM is a set of Matlab [1] routines for modelling beam dynamics in high energy particle accelerators. To develop an application, the user writes Matlab scripts that call classes and functions defined in SAMM. In other words, SAMM consists of a library from which an accelerator simulation may

be developed. The intentions behind SAMM are: for the physics to be as accurate as possible; for the code to be as short and simple as possible; and for the structure of the code to reflect as closely as possible the structure of a real accelerator. It is also intended that the code be structured to make it as easy as possible to extend its capabilities: that is, it should be straightforward for the user to add new beamline components, dynamical effects, and analysis procedures, as required.

Since simplicity and brevity are prioritised over performance, SAMM is unlikely to be the best code for large-scale simulations of complex accelerator systems. However, some effort has been made to make the code reasonably efficient, so for relatively simple systems, the speed of computation should be acceptable.

To achieve a logical and clear structure in SAMM, extensive use is made of object-oriented programming features in Matlab. Users are recommended to familiarise themselves with the section on “Object-Oriented Programming” in the Matlab documentation [?], before working with SAMM.

4 Dipole

sgsgjsgj

5 Quadrupole

sgjsgj

6 Sextupole

zgjzgjcv

7 Octupole

zxgjvxc

8 Multipole

xgvxcjg

9 RF Cavity

zxcgjuca

10 RF Accelerating Structure

zgjzcvcvz

11 NEW: Solenoid and RF Class

zcgcvcvzg

12 Orbit Corrector

zgjzvjjvzc

13 Screen

zdgjzcvhatd

14 Beam Position Monitor

zgjzcvzf

References

- [1] Mathworks, <http://www.mathworks.co.uk/>