

# APPLYING THE “SIMPLE ACCELERATOR MODELLING IN MATLAB” (SAMM) CODE TO HIGH LUMINOSITY LHC UPGRADE \*

K. M. Hock<sup>†</sup>, A. Wolski, University of Liverpool and Cockcroft Institute, UK  
R. B. Appleby, University of Manchester and Cockcroft Institute, UK

## Abstract

The “Simple Accelerator Modelling in Matlab” (SAMM) code is a set of Matlab routines for modelling beam dynamics in high energy particle accelerators. It includes a set of CUDA codes that can be run on a graphics processing unit. These can be called from SAMM and can potentially give a significant increase in tracking speed. To make use of this potential for the computationally intensive LHC upgrade simulations, we have developed additional Matlab and CUDA routines to simulate the full set of elements that are present in the Large Hadron Collider. We present the results of applying these codes to dynamic aperture calculations. These results are benchmarked against PTC and MADX.

## INTRODUCTION

Many codes are available for particle tracking, e.g. MADX, PTC [1], SixTrack [2], elegant [3], GPT [4], Zgoubi [5], etc. The “Simple Accelerator Modelling in Matlab” (SAMM) code [6] consists of a set of MATLAB classes and functions, with tracking routines also available in C and CUDA that can be called directly from MATLAB. SAMM was originally intended for educational use, but is now proving useful as a research tool. The tracking codes in SAMM are symplectic. Its tracking modules include drift, dipole, quadrupole, sextupole, octupole, RF cavity and solenoid. It also has Matlab routines for computing closed orbits, Twiss parameters of coupled beamline, etc. If an NVIDIA graphics processing unit (GPU) is installed on the computer, the CUDA codes can be called from Matlab and tracking computation will run on the GPU for greater speed.

In this paper, we report the following development:

- A Matlab code to read in MADX lattice file and convert it to SAMM format.
- Matlab, C and CUDA codes for tracking through a kicker and a multipole.
- Tracking through LHC and benchmarking against MADX and PTC.
- Computing of a dynamic aperture.

## METHOD AND CODES

We have developed a Matlab code to read into SAMM format a MADX sequence file for LHC optics version V6.500 [7]. To make it easier, we use MADX first to read

in the sequence file and then to output the data in table format to a text file. Each row of the table contains the parameters for each element. The main difference between the tracking functions in MADX and SAMM is in the dipole elements. In MADX, there are two types - RBEND is for a rectangular dipole field region and SBEND is for a sectorial one. In SAMM, there is only the sectorial dipole. So RBEND parameters must be converted to a sectorial dipole by changing the length of the magnet and the edge angles.

Tracking through dipole, quadrupole and solenoid in SAMM are based on expanding the Hamiltonian to second order in  $(x, p_x, y, p_y, z, \delta)$  and integrating the Hamilton's equations [8]. In particular, the dipole uses curvilinear coordinates so that  $(x, p_x, y, p_y, z, \delta)$  is expanded about a reference arc in the dipole field. Tracking through drift space uses an exact analytic expression. Tracking through sextupole and octupole make use of the drift-kick-drift method. In these ways, tracking in SAMM is fully symplectic.

In order to use SAMM for tracking in the LHC, we develop tracking codes for two additional elements - kicker and multipole. The kicker module is modified from the dipole module by changing the reference path from arc to straight line. This is done by setting the curvature of the curvilinear coordinates to zero. The multipole follows the drift-kick-drift method in the sextupole module. It replaces the kick by a sum of kicks from multipoles of different orders.

A main motivation in using SAMM for tracking in LHC is the availability of the CUDA code modules. (Another motivation is the features offered by Matlab.) If the computation can be done in parallel, running on a GPU can potentially be faster. If we need to track many particles through the same element, tracking of different particles can be done in parallel. This is implemented in the CUDA code for every element in SAMM. NVIDIA GPUs now support double precision arithmetic, but at a significant performance cost. The computation time could be more than twice as long. For the simulations in this paper, double precision is used.

In the dynamic aperture calculation that is discussed later, 5000 protons at 3.5 TeV are tracked for 10000 turns. We run this using the SAMM CUDA codes on an NVIDIA GTX580 GPU.

## BENCHMARKING

We start by benchmarking each element in SAMM against MADX and PTC [1], using parameters typical of

**05 Beam Dynamics and Electromagnetic Fields**

**D02 Non-linear Dynamics - Resonances, Tracking, Higher Order**

\* Work supported by Science and Technology Facilities Council, UK

<sup>†</sup> kmhock@liverpool.ac.uk

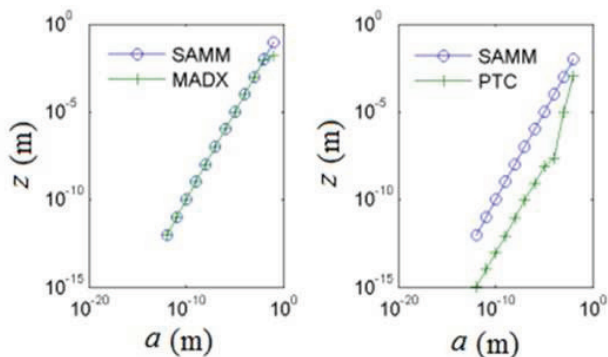


Figure 1: Benchmarking of a 10 m long quadrupole with  $k_1 = 0.01 \text{ m}^{-2}$  and a 100 GeV proton.

the element in the LHC. For example, we use a 10 m long quadrupole with normalised quadrupole gradient  $k_1 = 0.01 \text{ m}^{-2}$ . We use a comparatively low energy of 100 GeV so that off-axis particles would bend more. Since both SAMP and MADX use the paraxial approximation, bending further from the axis means that any difference is likely to be larger. This would give us a better idea of the possible errors. Using MADX, a 100 GeV proton at a certain offset from the axis is tracked for the length of the quadrupole. The offset is chosen to be  $(x, p_x, y, p_y, z, \delta) = (a \text{ m}, a, a \text{ m}, a, a \text{ m}, a)$ , where  $a$  is varied from  $10^{-12}$  to  $10^{-2}$ . The reason for this choice is to be able to vary all 6 coordinates in a simple way. The tracking starts at the entrance plane of the quadrupole and stops at the exit plane. At the exit plane, each coordinate is plotted against  $a$  on the same graph as the result from SAMP. As an example, the graph for coordinate  $z$  is shown in Fig. 1(a). Agreement is good. This comparison is repeated for PTC and shown in Fig. 1(b). Agreement is poor in this case. In another case, we track a 100 GeV proton through a kicker that bends the trajectory by 0.001 rad in the  $x$  direction and 0.001 rad in the  $y$  direction. The kicker is also rotated about the reference path by 0.001 rad. The same simulation as before is carried

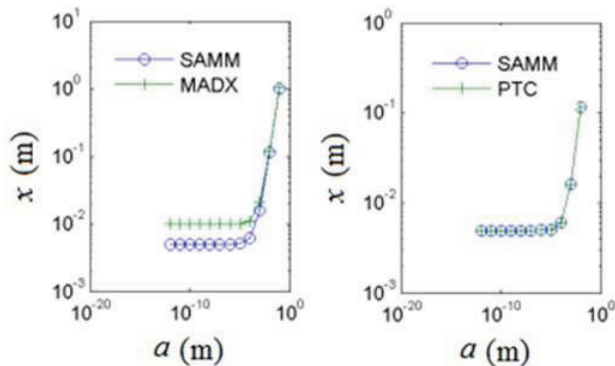


Figure 2: Benchmarking of 10 m long kicker with 0.001 rad for both horizontal and vertical kicks and a 100 GeV proton.

out. The comparison between MADX and SAMP for the  $x$  coordinate at the exit face of the kicker is shown in Fig. 2(a). The difference is clearly significant. The comparison between PTC and SAMP is shown in Fig. 2(b) and the agreement is good.

We have repeated this for the other elements. Agreement is better with MADX for some elements and coordinates, and better with PTC for others. Reasons for these disagreements are not clear, though a possible reason for MADX is that it uses the thin lens approximation. Overall, there seems to be better agreement with PTC in our test cases. Because of these differences, there is a need to further determine the reliability of SAMP. This we have done for elements where exact, analytic solutions are available. These include dipole, drift, solenoid and kicker, all of which have either constant or zero fields. We have generated the analytic solutions using a code based on a method that uses tiny arcs for tracking [9]. The results agree well with SAMP for all final coordinates of these 4 elements. With the LHC optics file converted from MADX to SAMP format as described in the previous section, we track a 3.5 TeV proton round the LHC for 100 turns. This is done using SAMP and PTC for the cases when  $\delta = 0$  and  $10^{-3}$ . The resulting trajectories in the  $(x, p_x)$  phase space at IP1 are shown in Fig. 3. There is good agreement between SAMP and PTC when  $\delta$  is 0. When  $\delta$  is  $10^{-3}$ , the trajectories still agree, but the actual points are different for higher turn numbers. The shape of the orbit in phase space is in very good agreement both on-energy and off-energy; but there seems to be some difference in the chromaticity between SAMP and PTC, i.e. there is a difference in the phase advance per turn for off-energy particles.

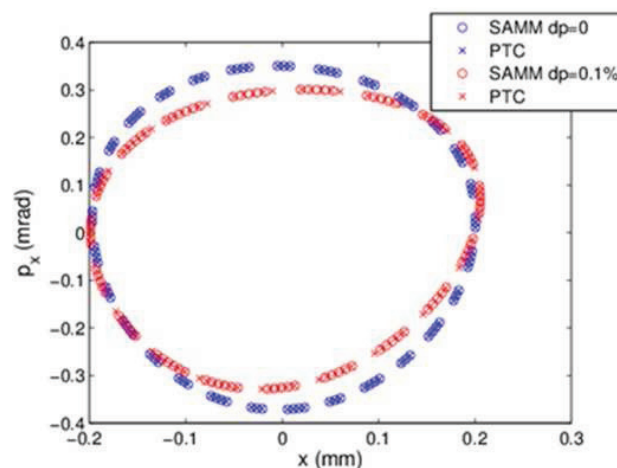


Figure 3: Benchmarking of  $(x, p_x)$  trajectory at IP1 by tracking a 3.5 TeV proton round the LHC for 100 turns.  $dp$  stands of  $\delta$ .

### DYNAMIC APERTURE

Using SAMP, we compute the LHC dynamic aperture on a GPU for the LHC optics version V6.500 [7].

We track a 3.5 TeV proton for 10000 turns around the LHC. We define a large physical aperture of 10 cm at every quadrupole and determine if the proton would hit the edge of one of the apertures. Collimator apertures are omitted in this simulation. The tracking is repeated for different values of initial  $(x, y)$  at the interaction point IP1. A uniform, 50 by 100 square grid of initial  $(x, y)$  is chosen. The size of this grid is chosen to be roughly twice the size of the chaotic boundaries (computed separately) in  $(x, p_x)$  and  $(y, p_y)$  phase spaces. Initial  $(p_x, p_y)$  are set to zeroes. Initial  $(x, y)$  for protons that hit an aperture edge are recorded and plotted on a graph. The result is shown in Fig. 4. Conventionally,  $(x, y)$  are given in units of  $\sigma_x$  and  $\sigma_y$  respectively.  $\sigma_x$  is the rms beamwidth in the  $x$  direction and is calculated using  $\sqrt{\varepsilon_x \beta_x}$ . The horizontal emittance  $\varepsilon_x$  is calculated from the normalised emittance  $\varepsilon_n$  using  $\sqrt{\varepsilon_n / \gamma}$ , where  $\gamma$  is the relativistic factor. Likewise for the  $y$  direction. The normalised emittance is the same in both horizontal and vertical directions and is given in the LHC design report [10] as 3.75 mm mrad. The beta functions  $\beta_x$  and  $\beta_y$  are computed using PTC and the values at IP1 where we determine the dynamic aperture are found to be 0.51 m and 0.57 m respectively.

Figure 4 is colour coded to show the number of turns before a proton is lost to an aperture. The gradual decrease in the area for higher turn numbers suggests that the dynamic aperture has converged with respect to turn number. This means that if we repeat the simulation using turn numbers larger than 10000, the result is likely to be similar. On the other hand, that result may still change for physical apertures larger than 10 cm. The simulation could be repeated for a few larger apertures to check for convergence.

The result in Figure 4 is significantly larger than published results [11]. This is because the LHC optics file that we have used has all multipole strengths set to zeroes. It is known that multipoles around the interaction regions would reduce the dynamic aperture [12].

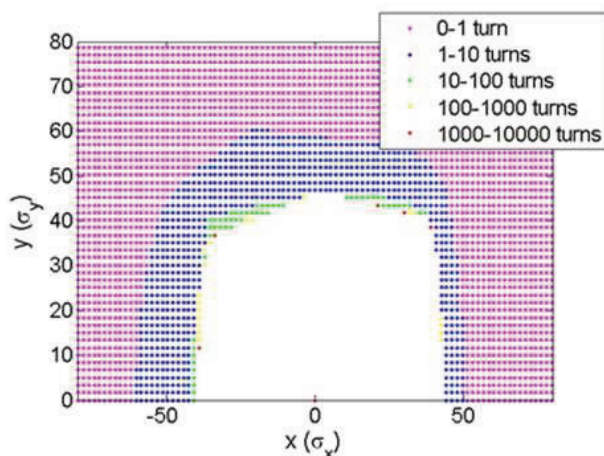


Figure 4: The empty region shows the initial  $(x, y)$  at IP1 of 3.5 TeV protons that are not lost after tracking for 10000 turns in the LHC.

## CONCLUSIONS

We have added Matlab, C and CUDA codes for kicker and multipole to SAMM. We have benchmarked these codes against MADX and PTC by tracking through each element and comparing the final coordinates. There is agreement in many but not all cases. Disagreements between MADX and PTC are significant in some cases. For magnetic elements with uniform fields - kicker, dipole, solenoid - we have benchmarked SAMM against analytic solutions and found good agreement in all cases. Finally, we have computed the dynamic aperture of the LHC on an NVIDIA GPU. Work is in progress to optimise the performance of the GPU.

## REFERENCES

- [1] MAD - Methodical Accelerator Design, <http://madx.web.cern.ch/madx/>
- [2] SixTrack – Single Particle Tracking Code, <http://frs.web.cern.ch/frs/>
- [3] elegant, [http://www.aps.anl.gov/Accelerator\\_Systems\\_Division/Accelerator\\_Operations\\_Physics/software.shtml#elegant](http://www.aps.anl.gov/Accelerator_Systems_Division/Accelerator_Operations_Physics/software.shtml#elegant)
- [4] GPT, <http://www.pulsar.nl/gpt/index.html>
- [5] Zgoubi, <http://sourceforge.net/projects/zgoubi/>
- [6] A. Wolski, Simple Accelerator Modelling in Matlab, [http://pcwww.liv.ac.uk/~awolski/main\\_links\\_computercodes.htm](http://pcwww.liv.ac.uk/~awolski/main_links_computercodes.htm)
- [7] LHC Optics Version V6.500, <http://proj-lhc-optics-web.web.cern.ch/proj-lhc-optics-web/V6.500/Quads.html>
- [8] A. Wolski, Linear Dynamics in Particle Accelerators Lectures 2 - 4, [http://pcwww.liv.ac.uk/~awolski/main\\_teaching\\_Cockcroft\\_LinearDynamics.htm](http://pcwww.liv.ac.uk/~awolski/main_teaching_Cockcroft_LinearDynamics.htm)
- [9] K. M. Hock and C. S. Edmonds, NIM A, 664, (2012), 140-147.
- [10] LHC Design Report Volume I, Chapter 2, Section 2.1, [https://edms.cern.ch/file/445830/5/Vol1\\_1\\_Chapter\\_2.pdf](https://edms.cern.ch/file/445830/5/Vol1_1_Chapter_2.pdf)
- [11] S. Fartoukh, M. Giovannozzi, NIM A, 671, (2012), 10-23.
- [12] M. Böge, et al, “Overview of the LHC dynamic aperture studies”, PAC 1997.