



FERMILAB-Conf-03/229 August 2003

## Simulation Aspects of Beam Collimation and Their Remedies in the MARS14 Code \*

M.A. Kostin, N.V. Mokhov, S.I. Striganov  
Fermilab, Batavia, IL 60510, USA

I.S. Tropin  
Tomsk Polytechnic University, Tomsk, 634034, Russia

August 7, 2003

### **Abstract**

Simulation aspects of beam collimation are described along with a number of tools and methods developed and used within the MARS14 framework. The tools and methods were implemented in order to relieve the burden of simulations needed for reliable calculations required for design of efficient collimation systems at high-intensity accelerators and colliders.

---

\*Presented paper at the *ICFA Workshop on Beam Halo Dynamics, Diagnostics, and Collimation (HALO'03)*, Montauk, Long Island, NY, May 19-23, 2003

# Simulation Aspects of Beam Collimation and Their Remedies in the MARS14 Code <sup>1</sup>

M.A. Kostin\*, N.V. Mokhov\*, S.I. Striganov\* and I.S. Tropin<sup>†</sup>

*\*Fermi National Accelerator Laboratory P.O. Box 500, Batavia, Illinois 60510*

*<sup>†</sup>Tomsk Polytechnic University, Tomsk, 634034, Russia*

**Abstract.** Simulation aspects of beam collimation are described along with a number of tools and methods developed and used within the MARS14 framework. The tools and methods were implemented in order to relieve the burden of simulations needed for reliable calculations required for design of efficient collimation systems at high-intensity accelerators and colliders.

## 1. INTRODUCTION

Collimators are an essential part of modern accelerators. With an increasing power of machines, the collimators have a major impact on beam, radiation shielding and backgrounds, which needs to be predicted and monitored. The predictions involve simulations with use of several models and requires precise description of physics processes. In this paper some of simulation aspects of beam collimation and related tools and methods designed and developed within the MARS14 framework [1] are addressed.

## 2. COUPLING ACCELERATOR AND RADIATION TRANSPORT CODES

Simulation of beam collimation systems normally consists of two stages. At the first stage, beam halo is transported through the accelerator lattice and lost particles are recorded. At the second stage, the lost particles are propagated through the material of collimators and other elements and their interactions with matter are simulated. There are currently no codes known to effectively perform both the stages within a single framework. Accelerator simulation codes that use a matrix formalism provide a very fast tracking of the particles in lattices but can not simulate interactions in matter. Vice versa, codes that can simulate the interactions implement slow step-wise

tracking algorithms. Therefore, at least two codes need to be employed. The variety of models implemented within various frameworks may lead to model inconsistencies, performance penalties and increased time needed for the development.

A MAD-MARS Beam Line Builder (MMBLB) [2] substantially improved recently [3] helps bind accelerator code models with ones implemented in MARS14. The MMBLB places elements of a beam line according to an optics file, created in a MAD style [4]. This format has been chosen given the fact that MAD is an industry standard. So that if an accelerator simulation code uses either a MAD input or optics file to represent a lattice then the model can easily be transferred to MARS14. The examples of recent use of the MMBLB include the Femilab Booster, Tevatron, Proton Driver, NuMI beam line, NLC and JPARC where beam loss distributions and induced radiation effects were studied and beam collimation systems were designed.

An optics file controls the longitudinal positions of elements along a beam line, element orientations and central magnetic field. An element is described with a single line. The first three fields are MAD keyword, type and name, respectively. Those serve for unambiguous identification of an element with a unique structure. The keyword can not be shortened, although this is a normal practice in MAD. This means that the keyword “KICK” can not be used in lieu of “KICKER”. There are two methods to identify the elements: 1) using the exact match of the type and name, 2) if this fails, then only the type is used. The next two fields in the optics file are the S-position (path length) of the element end and optical length expressed in meters. The following four parameters are related to the field. They represent the

---

<sup>1</sup> This work was supported by the Universities Research Association, Inc., under contract DE-AC02-76CH03000 with the U.S. Department of Energy.

magnet 'strength' in dipole, quadrupole, sextupole and octupole components, as defined in MAD, multiplied by the length of magnet. At this, the dipole parameter becomes the bend angle of a dipole. The last parameter is the roll angle of an element about the S-axis. The roll angle makes it possible to define a 3-D beam line with practically any orientation of its elements. The position and orientation of the elements are entirely governed by the rules of MAD [4].

There is no restriction on the number of beam lines in the new version of MMBLB. Each beam line is defined with its own optics file. Also, a beam line can be accommodated in an arbitrary place in the MARS global frame. The orientation of the first element of the beam line is specified by the user.

The MMBLB supports the full list of MAD elements except BEAMBEAM, MATRIX and LUMP. The parameters for elements represented with the keywords SOLENOID, HKICKER, VKICKER, KICKER, RFCAVITY, ELSEPARATOR, ECOILLIMATOR and RCOLLIMATOR can not be provided with an optics file. Instead, the parameters must be defined externally in the code. Whereas an optics file controls the longitudinal position of a beam line element, a user must provide several subroutines that describe the geometry, field, materials, volumes and names of sub-zones. In order to bind all those subroutines with a real element from the optics file they must be registered by means of a special subroutine.

The MMBLB offers a possibility to describe a simple beam enclosure. In order to do this, the enclosure must not change its geometry with respect to a reference orbit within some stretch. This method only works for flat 2-D beam lines. The code that describes the geometry of the beam enclosure must be provided by a user in a designated subroutine. Fig. 1 shows a NLC beam delivery section with a corresponding beam enclosure.

### 3. DESCRIPTION OF SCATTERING

A precise description of the scattering processes is exceptionally important for efficient calculations of collimation systems. Four parts are distinguished in a typical scattering spectrum (Fig. 2). With the increase of the momentum transfer  $t$ , they are respectively: multiple Coulomb scattering (MCS), elastic coherent and incoherent scattering on a nucleus and the scattering at larger angles due to diffractive processes. All four processes are carefully described and treated in the MARS14 code.

Multiple Coulomb scattering is most important for a description of scattering on an edge of collimators. Two methods are widely used: sampling from Moliere [5] and Gaussian [6] distributions. Limits of applicability of these approaches were determined in [7], where the

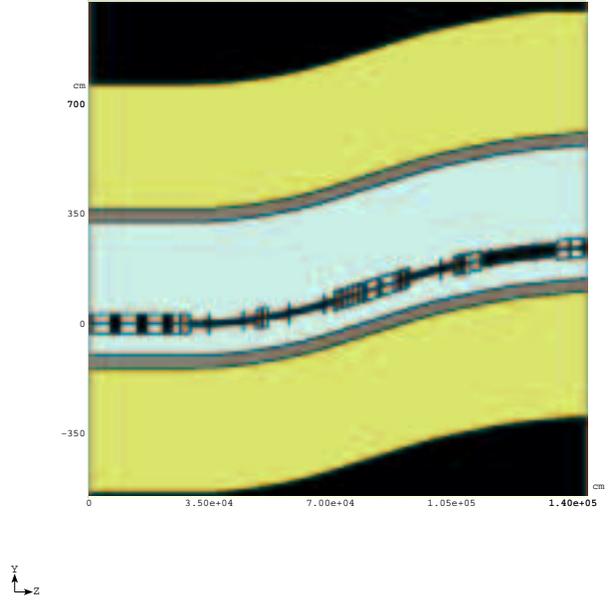


FIGURE 1. The NLC beam delivery section modeled with MMBLB.

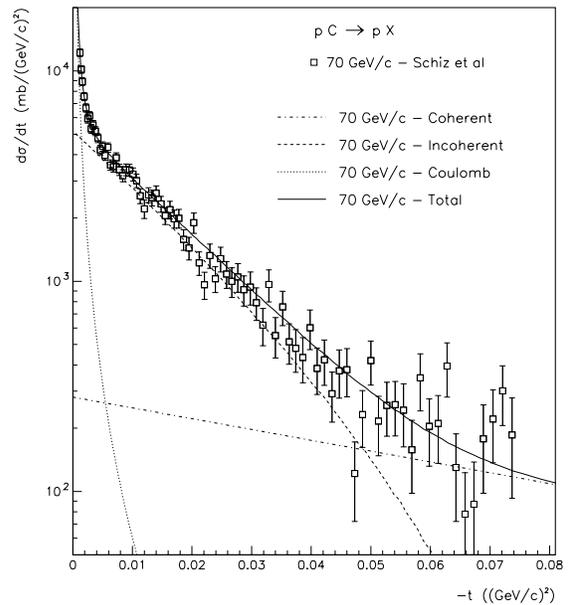
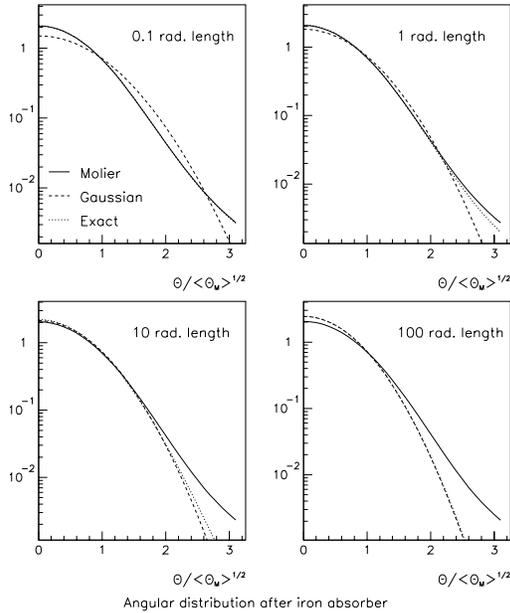


FIGURE 2. Proton-carbon scattering spectrum at 70 GeV/c.

Moliere theory of MCS was modified to take into account a nuclear screening. It was shown that the angular distribution obtained in such a way coincided with the Moliere one for the thicknesses of 0.1 - 1 radiation lengths and reached the Gaussian asymptotic for 100 - 1000 radiation lengths (Fig. 3). It was shown [8] that using Moliere Gaussian distributions alone can lead to

quite large errors in Monte-Carlo simulations especially at large angles. An efficient method to simulate (MCS) is based on a separate treatment of the "soft" and "hard" interactions. A large number of "soft" collisions is described using a "continuous scattering" approximation; a small number of "hard" collisions is simulated directly. A new analytical expression for a "continuous" angular distribution was recently developed [9]. A boundary angle between "soft" and "hard" collisions is determined as a function of a step-length providing a possibility for fast and precise simulation. Results of simulation using a Gaussian nuclear form-factor agree within 1% with analytical calculations [7], but the new algorithm provides a possibility to include an arbitrary charge distribution of projectile and target nucleus. A corresponding algorithm was implemented in MARS14 [9].



**FIGURE 3.** Angle distributions for particles scattered in iron absorbers of various lengths.

The coherent and incoherent elastic processes have smaller cross-sections compared to MCS but provide larger scattering angles. These processes may dominate for description of collimation systems that intercept beam particles scattered on a residual gas. The density of the residual gas is normally quite low, so that a noticeable angle deviation due to MCS is accumulated only in many turns. Coherent and incoherent elastic scattering are simulated in MARS14 using a fit to experimental data.

Description of diffractive dissociation in MARS14 is based on a triple-Reggeon phenomenology. A leading proton production cross section can be approximated by  $\exp(B_d t)/(1-x)$ , where  $x = p'/p$ , and  $p$  and  $p'$  are proton momenta before and after the interaction. The

exponential slope  $B_d$  is a factor of two larger than the slope of the incoherent elastic scattering. Due to non-marginal energy loss and a larger scattering angle, the diffractive scattering is unimportant for simulation of collimation systems, since in most cases, the particles scattered to large angles are lost in the lattice promptly and typically can not reach the collimators.

## 4. GEOMETRY DESCRIPTION AND VISUALIZATION

The collimation system development includes thorough radiation transport and energy deposition simulations in beam elements, shielding and around to meet regulatory requirements on prompt and residual radiation in and around the tunnel. A quite comprehensive and complex geometry, materials and magnetic field descriptions are needed in order to address all these features. The model complexity can cause errors. Two modules in MARS14 help deal with this problem.

The first one allows a choice of geometry description. Depending on configuration and user's experience, one can use one of the geometry packages available in MARS14: Standard ( $r$ - $z$ - $\phi$ ), MCNP, Extended and Non-Standard (arbitrary user-defined) [1]. Any part of the system, described within the first three options, can be overwritten by the Non-Standard geometry segments. In the MCNP mode, the code uses input geometry description in the MCNP format [10] (except lattices and universes). The term "Extended" refers to an extension beyond the Standard zones. In the current version, Extended geometry uses a combination of boxes, cylinders, spheres, cones and tetrahedra, similar to the methods used by other Monte Carlo programs, particularly GEANT [11]. These elements can be arbitrary positioned and rotated by a set of transformation matrices, and divided to sub-zones. For example, a box can be sub-divided along all three directions. The elements can overlap.

The second module is a powerful Graphical User Interfaces (GUI), used for visualization and debugging of geometry, materials and magnetic field descriptions, simulated processes and calculated results. This user-friendly MARS-GUI-SLICE tool is absolutely vital in serious studies of accelerator, detector and shielding applications and used with enthusiasm by the MARS14 community worldwide.

A new 3-D extension has recently been developed and added to the MARS14 code system [12]. It is built on the top of the OPENINVENTOR [13] SGI library. A 3-D view panel is launched from the MARS-GUI-SLICE window (Fig. 4). One can use the OPENINVENTOR 3-D GUI to rotate, zoom in and out and move the scene. Beside those standard OPENINVENTOR functions, other

useful features have been added such as changing from a solid to wire-frame rendering, a light editor, ability to save the scene image to a postscript file and modify the background. One can also plot 3-D particle tracks and modify a color and transparency of materials chosen from the MARS-GUI-SLICE window.

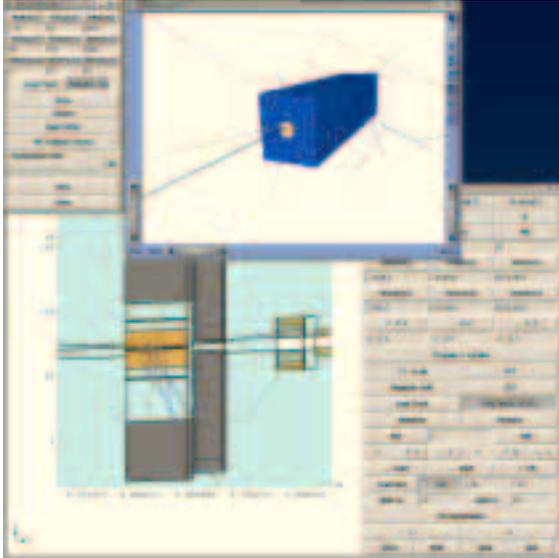


FIGURE 4. MARS14-3D-GUI example.

## 5. TRACKING ACCURACY

There are several particle tracking issues vital in collimation applications. All of them are treated as precise and efficient as possible in the MARS14 code. These include:

- *Boundary localization.* It is done exactly in the Extended geometry mode, and controlled by the region-material depended parameter to the accuracy required in the given application. Special options are provided in the code for efficient simulations in a system which includes, for example, tiny objects such as vacuum windows, beam pipes, electrostatic septum wires etc (microns to millimeters), implemented into extended vacuum voids (hundreds meters), followed by the bulk shielding (meters to kilometers).

- *Edge scattering.* This is especially tough problem for charged particle tracking near the vacuum-material boundaries in presence of magnetic field and quasi-continuous processes such as Coulomb scattering.

- *Tracking in magnetic field.* A step-size is chosen according to the field value in the plane perpendicular to the direction of motion, particle momentum and its ratio to the primary beam momentum, with a possible user control on top of that. MARS14 algorithms provide typically a ten micron (microradian) accuracy for the beam line and accelerator systems of several kilometers long.

## 6. CONCLUSIONS

MARS14 is the powerful code that offers a number of tools and methods for simulating collimation systems. The recent advancements, described in this paper, include the MAD-MARS Beam Line Builder, a careful treatment of the scattering processes, several options for geometry description, Graphical User Interface and high tracking accuracy. Two types of simulation that require the most significant computational time are precise tracking in magnetic field of very long beam lines and accelerator lattices and a deep penetration problem in bulk shielding around collimators. A normal approach to speed the calculations up is the use of a number of relatively short computational jobs and then average the results. A paradox arises however: in order to obtain a statistically significant result by averaging the results of the shorter runs, the shorter job outputs have to be statistically significant already. Otherwise, the use of the averaging procedure that assumes the values distributed over Gaussian, would lead to biased both the errors and mean values. A simple way out of this is to use parallel computing. A corresponding work with MARS14 is underway.

## REFERENCES

1. N.V. Mokhov, "The MARS Code System User's Guide", Fermilab-FN-628 (1995); N.V. Mokhov, O.E. Krivosheev, "MARS Code Status", Proc. Monte Carlo 2000 Conf., p. 943, Lisbon, October 23-26, 2000; Fermilab-Conf-00/181 (2000); <http://www-ap.fnal.gov/MARS/>.
2. O. Krivosheev et al., "A Lex-based MAD parser and its applications," FERMILAB-CONF-01-142-T; Proc. IEEE Particle Accelerator Conference (PAC 2001), Chicago, Illinois, 18-22 Jun 2001.
3. M.A. Kostin, O.E. Krivosheev, N.V. Mokhov, I.S. Tropin, "An Improved MAD-MARS Beam Line Builder: User's Guide", FERMILAB-FN-738, 2003.
4. H. Grote, F.C. Iselin, "The MAD program (Methodical Accelerator Design)", CERN/SL/90-13.
5. G.Z. Molier, Z. Naturforsch 2a, p.133, 1947; Z. Naturforsch 3a, p.78, 1948.
6. B. Rossi and K. Greisen, Rev.Mod.Phys 13, p.240, 1941.
7. I.S. Baishev, N.V. Mokhov and S.I. Striganov, Sov. J. Nucl. Phys. 42, p.745, 1985.
8. S.I. Striganov, Nucl.Phys. B (Proc. Suppl) 51A, p.172, 1996.
9. N.V. Mokhov and S.I. Striganov, LA-UR-03-4262.
10. <http://laws.lanl.gov/x5/MCNP>
11. <http://wwwinfo.cern.ch/asd>.
12. J.P. Rzepecki, M.A. Kostin and N.V. Mokhov, "3D Visualization for the MARS14 Code", FERMILAB-TM-2197, Jan. 2003.
13. <http://www.openinventor.com>.