1st International Workshop on Space Radiation Research and
11th Annual NASA Space Radiation Health Investigators' Workshop
Arona (Italy), May 27-31, 2000

Development of a space radiation Monte Carlo computer simulation based on the FLUKA and ROOT codes

L.S. Pinsky[1], T.L. Wilson[2], A. Ferrari[3], P. Sala[4], F. Carminati[5], R. Brun[6]

1. *University of Houston, Houston TX (USA)*
2. *NASA, Johnson Space Flight Center Houston TX (USA)*
3. *INFN, Milan (Italy)*
4. *CERN, Geneva (Switzerland)*

Abstract

This NASA funded project is proceeding to develop a Monte Carlo-based computer simulation of the radiation environment in space. With actual funding only initially in place at the end of May 2000, the study is still in the early stage of development. The general tasks have been identified and personnel have been selected.

The code to be assembled will be based upon two major existing software packages. The radiation transport simulation will be accomplished by updating the FLUKA Monte Carlo program, and the user interface will employ the ROOT software being developed at CERN. The end-product will be a Monte Carlo-based code which will complement the existing analytic codes such as BRYNTRN/HZETRN presently used by NASA to evaluate the effects of radiation shielding in space. The planned code will possess the ability to evaluate the radiation environment for spacecraft and habitats in Earth orbit, in interplanetary space, on the lunar surface, or on a planetary surface such as Mars. Furthermore, it will be useful in the design and analysis of experiments such as ACCESS (Advanced Cosmic-ray Composition Experiment for Space Station), which is an Office of Space Science payload currently under evaluation for deployment on the International Space Station (ISS).

FLUKA will be significantly improved and tailored for use in simulating space radiation in four ways. First, the additional physics not presently within the code that is necessary to simulate the problems of interest, namely the heavy ion inelastic processes, will be incorporated. Second, the internal geometry package will be replaced with one that will substantially increase the calculation speed as well as simplify the data input task. Third, default incident flux packages that include all of the different space radiation sources of interest will be included. Finally, the user interface and internal data structure will be melded together with ROOT, the object-oriented data analysis infrastructure system. Beyond the benefits of 'objectivity', ROOT's incorporation will also provide a graphical user interface with powerful tools for input prior to the calculation, as well as for data analysis and visualization of the results.

KEYWORDS: Cosmic rays, Monte Carlo, transport code, simulation software.

1. Introduction

The radiation environment in space is a complex problem to model. Trying to extrapolate the projections of that environment into all areas of the internal spacecraft geometry is even more daunting. With the support of our CERN, INFN and NASA colleagues, our research group at the University of Houston is embarking on a project to develop a radiation transport tool that is tailored to the problem of taking the external radiation flux incident on any particular spacecraft and simulating the evolution of that flux through a geometrically accurate model of the spacecraft material. The output will be a prediction of the detailed nature of the resulting internal radiation environment within the spacecraft as well as its secondary albedo. We are also planning to incorporate accurate human phantoms available for insertion into the geometry volumes. Beyond doing the physics transport of the incident flux, the software tool we are developing will provide a self-contained stand-alone object-oriented analysis and visualization infrastructure. It will include a graphical user interface and a set of input tools to facilitate the simulation of space missions in terms of nominal radiation models and mission trajectory profiles.

The goal of this study is to produce a code that is considerably more accurate and user-friendly than existing Monte Carlo-based tools for the evaluation of the space radiation environment. Furthermore, the code will be a useful complement to the currently existing analytic codes in the BRYNTRN/HZETRN family for the evaluation of radiation shielding [1]. The new code will be directly applicable to the simulation of environments in low Earth orbit, on the lunar surface, on planetary surfaces (including the Earth) and in the interplanetary medium such as on a transit to Mars (and even in the interstellar medium). The new code is to be based upon the melding together of two existing evolving codes, FLUKA [2-10], a Monte Carlo radiation transport program, and the object-oriented physics analysis infrastructure known as ROOT [11-12]. The software will include modules whose underlying physics base can continue to be enhanced and updated for content, as future data become available beyond the timeframe of the initial version now foreseen. This future maintenance will be available from the authors of FLUKA as part of their continuing efforts to support the users of the FLUKA code within the particle physics community. In keeping with the spirit of developing an evolving physics code, we are planning as part of this project to participate in the efforts to validate the core FLUKA physics in ground-based accelerator test runs. The emphasis of these test runs will be the physics of greatest interest in the simulation of the space radiation environment.

Such a tool will be of great value to planners, designers and operators of future space missions, as well as for the design of the vehicles and habitats to be used on such missions. It will also be of aid to future experiments of various kinds that may be affected at some level by the ambient radiation environment, or in the analysis of hybrid experiment designs that have been discussed for space-based astronomy and astrophysics. The tool will be of value to the Life Sciences personnel involved in the prediction and measurement of radiation doses experienced by the crewmembers on such missions. In addition, the tool will be of great use to the planners of experiments to measure and evaluate the space radiation environment itself. It can likewise be useful in the analysis of safe havens, hazard migration plans, and in modeling the radiation exposure of electronic circuits to evaluate the probability of Single Event Upsets (SEU). This code will provide an important complimentary check on the predictions of analytic codes such as BRYNTRN/HZETRN that are presently used for many similar applications, and which have shortcomings that are more easily overcome with Monte Carlo type simulations. Finally, it is our intention to make the new software package modular and sufficiently flexible to allow the parallel use of FLUKA, GEANT 3.21 or GEANT4 as the physics transport engine.

These proceedings include descriptions of FLUKA and the other Monte Carlo codes[13], so this paper will not attempt to recover that ground. Rather, only the needed modifications to FLUKA will be described along with a discussion of the interface that will be developed. At present there exists a prototype for the general concept of employing the ROOT infrastructure as a Monte Carlo interface. The software team from the ALICE experiment that is currently being designed and constructed for deployment at the new LHC facility at CERN is developing a software package known as ALIROOT. The core of ALIROOT is a Virtual Monte Carlo Interface (VIRMCI). The general features of this VIRMCI approach will be described here along with a discussion of the planned features of the final version of the new software package that is the subject of the project being presented in this paper.

2. Planned FLUKA Modifications

Given FLUKA's comparative excellence as a radiation transport code, the question arises as to what modifications are required to make it useful for the routine simulation of the space radiation environment? First, although the FLUKA team is in the process of adding the complete range of heavy-ion interactions needed to simulate the propagation of the heavy cosmic rays, it is not their highest priority, and assistance is needed to facilitate that implementation. This task will be broken into two parts. The simulation of heavy-ion inelastic collisions with traversed nuclei above about 5 GeV/n will be accomplished by incorporating a Dual-Parton model (DPMJET) [14]. The simulation of these interactions below 5 GeV is more problematic and will require a considerable effort. In the short run we plan to develop an interim version of the pre-equilibrium code, PEANUT [4], which has been successfully employed in the current version of FLUKA to simulate the proton-nucleus inelastic collisions. At the same time we plan to participate in a collaborative effort with other groups in the development of more accurate models based upon a careful assessment of all of the existing data. Other physics improvements will be incorporated in the core physics code by the FLUKA team as they become available. Such enhancements will also continue to occur beyond the time of the initial public distribution of the new software.

Secondly, since the FLUKA code was first designed there has been substantial progress in the development of 3-D geometry software. Driven by computer graphics technology and ray-tracing applications, many new 3-D geometry packages are available. There is considerable room for improvement in the present geometry package used in FLUKA. The benefits of such an improvement would include an increase in the calculation speed during propagation simulations, and the additional bonus of easy access to industry-standard input and graphic display formats. As described below in Section 4, this task is a relatively easy one, once the optimum geometry package is identified.

Beyond the need for some physics additions and internal geometry enhancements, however, the major reason that FLUKA is not more widely used is the awkward nature of the present interface to the code. Admittedly, the current input format is archaic and somewhat limited. Furthermore, the tools available to structure the form of the output while providing significant capabilities, are also limited and difficult to use. To address these limitations, the thrust of our present study is to meld FLUKA together with the recently released physics analysis infrastructure software known as ROOT. This task will require an intimate conversion of the FLUKA data structures as opposed to the simple serial attachment of one code to another.

3. ROOT, A Data Analysis Infrastructure

ROOT, which is rapidly becoming a major force in the world of particle physics analysis software, is the product of Dr. Rene Brun of CERN [11]. At its heart, ROOT is based upon Object-Oriented (OO) data structures. This use of OO programming allows many difficult tasks with multiple uses to be done only once. For example, after the task of inputting the complex geometry of the ISS is done, that same information can be used seamlessly as the input for the

Monte Carlo transport calculations. The same data structure can then be applied to visualizations of individual Monte Carlo events in a fly-through 3-D event viewer, or for choosing regions to provide plots of individual summed quantities of interest. In fact it can be employed for any analysis or display uses that need to specify or depict some portion of, or indeed the entire structure in question. The best feature of ROOT is that much of the manipulation is provided via GUI (Graphical User Interface) menus and displays. Further, when special features need to be added, ROOT uses C++ as a scripting language. This implementation of the scripting language allows the user to create structures and functions that blend seamlessly into ROOT, providing a real user-extension of the system. Moreover an intelligent pre-processor completely solves the problem of persistency of objects in disk storage. ROOT features and examples can be viewed on the Web at http://root.cern.ch. That site contains the downloadable software with accompanying tutorials.

Within C++, one is allowed to define transportable data structure objects called classes. One advantage of adopting ROOT is that the large and growing community of users is continually developing new classes. These classes are distributed in an open source model similar to that employed by the developers of the Linux operating system. This allows users rapidly to incorporate the useful developments of others into their codes. As different groups begin to use this new software for space radiation simulation, it is anticipated that an exchange of well-documented libraries containing such additions will become common.

ROOT is a software package whose scope and capability is much more easily understood by example rather than by a textual description. The following section describes ALIROOT, which is an example of an adaptation of ROOT to control and evaluate Monte Carlo simulations of the ALICE experiment. This existing code will serve as a starting point for the new space radiation simulation code.

4. ALIROOT and the Virtual Monte Carlo Intrerface

Given ROOT's strengths as a data analysis infrastructure, it is almost natural to consider developing a general Monte Carlo radiation transport interface. One can break down the general problem of simulating the transport of radiation through a complex geometry into six basic steps:

i) Inputting the geometry including the specification of the material composition of each sub-volume;
ii) Inputting the incident flux form;
iii) Setting up the desired scoring;
iv) Initializing the transport code as required for the materials and options to be used;
v) Specifying the number of incident particles to be included in the run;
vi) Executing an event transport cycle for each incident particle during which the specified scoring is done as required.

The information needed by the last cycle, the transport cycle, from the earlier input steps is typically included in one or more files. ROOT is ideally designed to allow the construction of GUI-based tools to accomplish these steps. It is in the last step that the features of the Virtual Monte Carlo Interface come into play. During the transport process, as particles move from one sub-volume to another, interactions are determined by the reference to known cross sections appropriate for the composition of each sub-volume. When secondary particles are produced in these interactions, they are pushed onto a stack for subsequent transport after the original incident primary is followed to its eventual end. In each sub-volume, any desired scoring could be done. For example the total energy deposited in that sub-volume by the current particle can be calculated and added to the output data bin for that sub-volume.

The VIRMCI requires that the transport code have the ability to break out at each- point where scoring needs to be accomplished to allow that scoring to be done in ROOT data structures. This process can be facilitated if the core data structures of the transport program itself can be mapped directly into ROOT data structures. In the case where the Monte Carlo code is written in FORTRAN, as is FLUKA, utilities exist to enable the direct conversion of FORTRAN common blocks into C++ classes. This means a great saving in execution speed because the internal Monte Carlo values do not have to be passed back to the ROOT-based routines as arguments. Rather, the ROOT-based code will have direct access to any of the transport code's data structures for use in scoring. Thus, one can employ the FLUKA transport code to simulate the physics, and use ROOT-based code to setup the needed FLUKA input files as well as to score the parameters of interest during the simulation.

In FLUKA, it turns out that the geometry is also dealt with in a sufficiently modular way that it too can be broken out and dealt with externally to do the physics calculations. This is because at each spatial step, FLUKA enters a specific set of routines that simply advise the physics calculations what material the present sub-volume is composed of and how far the current particle can go in that sub-volume, given its direction of travel, before its trajectory intersects a boundary of that sub-volume. This allows one to substitute an external geometry package for the internally provided FLUKA geometry package. Currently, such substitute packages exist that allow FLUKA to be run using GEANT4 geometry files as inputs. Since translators exist to convert GEANT 3.21 geometry into GEANT4 geometry, one can already effectively choose to run FLUKA with any of these geometries. ALIROOT takes advantage of these capabilities, and indeed, we hope to be able to explore the possibility of a new geometry package that is optimized for particle transport to increase substantially the execution speed of the code.

In the current version of ALIROOT, the major effort is directed at the creation of generic scoring capabilities. Little attention has gone into input interfacing. Typically, one still edits the original input file with a text editor and then specifies that file through a ROOT macro. Our intention is to add the capability to edit the contents of the required input files directly from ROOT GUI menus and dialog boxes. This will free the user from having to learn the input file syntax.

Although not strictly part of the Monte Carlo codes themselves, analysis and display of the data are another strong suit of ALIROOT. Event displays can plot depictions of the trajectories superimposed on the geometry. Fully 3-D rotatable images that can be zoomed and explored in detail allow for strong visual analysis tools. Conventional histogramming where the plotted events can be subjected to any number of filters is also routinely possible. These and other tools are planned for the new code.

5. The Goals of the Current Project

The goal of this study is to produce a well-documented code that is useful for simulation of the space radiation environment, which can be widely distributed and employed by a variety of users. It is anticipated that the core project will be completed within three years. It is our goal to distribute a stable version of the entire new package with all of the anticipated features being functional.

In addition to the development of the code itself, databases will also be included to allow ease of simulation of the ambient radiation environment in space. Models of the trapped radiation, the galactic cosmic radiation, the modulation effects of the solar cycle, the albedo from the Earth's atmosphere, and a variety of solar flare examples will be included. Tools will be provided to allow users to tailor the inputs to specific mission profiles. That is, in addition to specifying the choice of event generators, it is planned to provide a ROOT-based infrastructure that will allow the user to choose the inputs by specifying the trajectory or orbital parameters and the mission dates. The program will then select the input spectra based on the included models that best represent the anticipated environment.

We are proceeding to modify FLUKA to include heavy-ion interactions and as needed for use with ROOT inputs. The result will be a combined package that will allow the use of the ROOT interface to input to and control FLUKA, and then seamlessly continue within ROOT to do the analysis of the output from FLUKA. Our approach will be generic, following the Virtual Monte Carlo Interface, and it will allow the developed infrastructure to be readily adapted to the use of any Monte Carlo transport code that allows the required internal access.. Such a package will be useful not only for the final analysis of the eventual end product output, but it will be invaluable for iterative development of the simulation itself. Integrated single-event displays and the ability to examine the performance of the simulation in geometric as well as physics detail, will be of great benefit in providing rapid and efficient application of the code to the variety of tasks for which it is being developed. The analysis software can even be employed within the ROOT framework to create, fill and plot histograms during the actual data taking, allowing one to perform classical statistical analysis on them, including multi-parametric fits in real-time.

REFERENCES

[1] Wilson JW et al. Transport Methods and Interactions for Space Radiation. JW NASA Ref Pub 1257, 1991.

[2] Aarnio P et al. Proc. of the MC93 Int Conf on Monte Carlo Simulation in High-Energy and Nuclear Physics, 22-26 Feb, 1993. Dragovitsch P, Linn S and Burbank M Eds. Singapore. World Scientific, 88 1994.

[3] Ajaltouni Z et al. [ATLAS Collaboration]. Results From a Combined Test of an Electromagnetic Liquid Argon Calorimeter with a Hadronic Scintillating-Tile Calorimeter. CERN-PPE/96-178, Nov 8, 1996.

[4] Ferrari A, Sala P. GEANT Hadronic Event Generators: A Comparison at the Single Event Level. ATLAS Note Phys-No-086. June 6, 1996. CERN, Geneva (Switzerland) 1996.

[5] Battistoni G, Ferrari A, Forti C, Scapparone E. Simulation of Muon Transport at High Energy: Comparison of a Few Different Codes. Nucl Inst and Meth 1997: A394; 136.

[6] Fasso A et al. Proc. of the Workshop on Simulating Accelerator Radiation Environment. Santa Fe, Jan 11-15, 1993; 134 (available as Los Alamos Report LA-12835-C, 1994).

[7] Fasso A, Ferrari A, Ranft J, Sala P. New Developments in FLUKA Modelling of Hadronic and EM Interactions. Proc 3rd Workshop on Simulating Accelerator Radiation Environment. KEK, Tsukuba (Japan), May 7-9, 1997.

[8] Ferrari A, Rancati T, Sala P. FLUKA Applications in High Energy Problems: From LHC to ICARUS and Atmospheric Showers. Proc 3rd Workshop on Simulating Accelerator Radiation Environment. KEK, Tsukuba (Japan), May 7-9, 1997.

[9] Ferrari A, Sala P. Intermediate and High Energy Models in FLUKA: Improvements, Benchmarks and Applications, Proc Int Conf On Nucl Data for Science and Tech [NDST-97], Trieste (Italy), May 19-24, 1997.

[10] Sala P, Ferrari A. Proc. of the MC93 Int. Conf. on Monte Carlo Simulation in High-Energy and Nuclear Physics, Feb 22-26, 1993. Dragovitsch P, Linn S, Burbank M Eds. Singapore. World Scientific 1994; 277.

[11] Brun R, et al. Proc of CHEP 97 (Computing in High Energy Physics), Berlin, April 7-11 1997 (in press, Elsevier 1997).

[12] Brun R, et al. ROOT Overview and ROOT Tutorials, CERN, Geneva (Switzerland), Available directly from the ROOT web site at http://root.cern.ch, 2000.

[13] Ferrari A, Sala P. The FLUKA radiation transport code and its use for space applications. Phys Med 2001: XVII(Supplement 1); 72-80.

[14] Ranft J, Ritter S. Z Phys 1985: C27; 413.